



TUGAS AKHIR - TE 145561

**RANCANG BANGUN ALAT PEMBERI PAKAN IKAN OTOMATIS
PADA KOLAM IKAN GURAMI BERBASIS ARDUINO**

Fastabiq Khoir Alblitary
NRP 2213030038

Dosen Pembimbing
Ir. Joko Susila, M.T.
Muhtadin, ST. MT.

PROGRAM STUDI KOMPUTER KONTROL
Departemen Teknik Elektro Otomasi
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017



FINAL PROJECT - TE 145561

DESIGN OF AUTOMATIC FISH FEEDER ON GURAMI POND USING ARDUINO BASED CONTROLLER

Fastabiq Khoir Alblitary
NRP 2213030038

Advisor
Ir. Joko Susila, M.T.
Muhtadin, ST. MT.

COMPUTER CONTROL STUDY PROGRAM
Electrical and Automation Engineering Department
Vocational Faculty
Institut Teknologi Sepuluh Nopember
Surabaya

2017

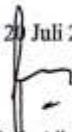
PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul **"Rancang Bangun Alat Pemberi Pakan Ikan Otomatis Pada Kolam Ikan Gurami Berbasis Arduino"** adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, 20 Juli 2017



Fastabiq Khoir Alblitary
NRP 2214030038

-----Halaman ini sengaja dikosongkan-----

**RANCANG BANGUN ALAT PEMBERI PAKAN IKAN
OTOMATIS PADA KOLAM IKAN GURAMI BERBASIS
ARDUINO**

TUGAS AKHIR

Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Ahli Madya Teknik
Pada

Program Studi Komputer Kontrol
Departemen Teknik Elektro Otomasi
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember

Menyetujui :

Dosen Pembimbing I

Dosen Pembimbing II

Ir. Joko Susilar M.T.
NIP. 1966 06 05 1991 02 1001

Muhtadin, ST, M.T.
NIP. 1981 06 09 2009 12 1003

**SURABAYA
JULI, 2017**

-----Halaman ini sengaja dikosongkan-----

RANCANG BANGUN ALAT PEMBERI PAKAN IKAN OTOMATIS PADA KOLAM IKAN GURAMI BERBASIS ARDUINO

Nama : Fastabiq khoir Alblitary
Pembimbing I : Ir. Joko Susila, M.T.
Pembimbing II : Muhtadin, ST. MT.

ABSTRAK

Permasalahan yang umum dirasakan petani ikan gurami adalah bagaimana meningkatkan efektivitas usaha agar dapat meningkatkan kesejahteraan petani gurami. Salah satunya adalah bagaimana melakukan pemberian pakan ikan secara rutin dan terjadwal tanpa menggunakan tenaga manusia untuk menghemat ongkos budidaya. Proses manual saat ini yaitu dengan menaburkan pakan ke permukaan kolam ketika masuk waktu makan bagi ikan. Apabila pemilik kolam berhalangan mengurus kolamnya maka pertumbuhan dan kesehatan ikan dapat terhambat.

Sebagai solusi permasalahan tersebut, diperlukan alat yang dapat mengotomatisasi pemberian pakan ikan, yang dapat berjalan secara terjadwal sehingga dapat membantu mempermudah budidaya ikan tanpa bergantung pada tenaga manusia.

Hasil dari penelitian ini adalah terciptanya sebuah alat pemberi pakan ikan otomatis untuk membantu menyelesaikan masalah pemberian pakan pada pembudidayaan ikan gurami. Sehingga pembudidayaan menjadi lebih efisien dan para petani ikan menjadi lebih sejahtera, serta dapat membantu meningkatkan produksi pangan nasional.

Kata Kunci : Arduino, Motor *DC*, Potensiometer

-----Halaman ini sengaja dikosongkan-----

DESIGN OF AUTOMATIC FISH FEEDER ON GOURAMI FISH POND USING ARDUINO BASED CONTROLLER

Name : Fastabiq khoir Alblitary
Advisor I : Ir. Joko Susila, M.T.
Advisor II : Muhtadin, ST. MT.

ABSTRACT

A common problem that is encountered by gourami fisheries is how to increase the productivity of their business so that their prosperity increased. One of the problem is how to feed the fish automatically and scheduled periodically without human power to economize fish raising cost. Current manual feeding process is by sowing the fish food on the pond's surface when fish feeding time came. If the fish raiser/pond owner cannot look after his pond, then the fish growth could be slowed.

Therefore, as a solution to this problem, we need a machine to automate the fish feeding and water circulation that can run periodically and scheduled so that the gurami fisheries could run without depending on human resources.

The results of this final project is an automatic fish feeder system to help solving the problem on gurami fisheries. Finally, we hope that the fish raising process became more efficient and the fish raiser prosperity is increased, and also able to help increasing national food production.

Keywords : Arduino, DC Motor , Potentiometer

-----Halaman ini sengaja dikosongkan-----

KATA PENGANTAR

Puji syukur penulis panjatkan kehadiran Allah SWT yang selalu memberikan rahmat dan hidayah-Nya sehingga Tugas Akhir ini dapat terselesaikan dengan baik. Shalawat serta salam semoga selalu dilimpahkan kepada Rasulullah Muhammad SAW, keluarga, sahabat, dan umat muslim yang senantiasa meneladani beliau.

Tugas Akhir ini disusun untuk memenuhi sebagian persyaratan guna menyelesaikan pendidikan Diploma-3 pada Program Studi Komputer Kontrol, Departemen Teknik Elektro Otomasi, Fakultas Vokasi, Institut Teknologi Sepuluh Nopember Surabaya dengan judul:

RANCANG BANGUN ALAT PEMBERI PAKAN IKAN OTOMATIS PADA KOLAM IKAN GURAMI BERBASIS ARDUINO

Dalam Tugas Akhir ini, dirancang alat pemberi makan ikan berdasarkan waktu makan ikan yang telah diatur sebelumnya oleh pengguna.

Penyusun mengucapkan terimakasih utamanya kepada kedua orang tua yang mana dukungan moril dan doa beliau berdua sangat berarti, serta semua pihak yang telah mendukung dalam pembuatan buku tugas akhir ini, termasuk dosen pembimbing kami, yaitu Bapak Ir. Joko Susila, M.T. dan Bapak Muhtadin, ST. MT. yang telah menyempatkan waktunya untuk membimbing kami dalam menyelesaikan tugas akhir ini. Salam sukses tidak lupa penulis ucapkan pada partner tugas akhir Budris Ariwibowo, Om Wawan dan pasukan Kendangsari, serta teman-teman angkatan 2014 atas bantuan dan dukungan semangatnya serta semua pihak yang telah membantu dan tidak dapat penulis sebutkan satu persatu.

Penulis menyadari dan memohon maaf atas segala kekurangan pada Tugas Akhir ini. Akhir kata, semoga Tugas Akhir ini dapat bermanfaat dalam pengembangan keilmuan di kemudian hari.

Surabaya, 20 Juli 2017



Penulis

-----Halaman ini sengaja dikosongkan-----

DAFTAR ISI

	HALAMAN
HALAMAN JUDUL.....	i
HALAMAN JUDUL.....	i
PERNYATAAN KEASLIAN TUGAS AKHIR	v
HALAMAN PENGESAHAN	vii
ABSTRAK	ix
ABSTRACT	xi
KATA PENGANTAR.....	xiii
DAFTAR ISI.....	xv
DAFTAR GAMBAR	xix
DAFTAR TABEL	xxiii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Permasalahan	1
1.3 Batasan Masalah.....	2
1.4 Tujuan	2
1.5 Metodologi Penelitian	2
1.6 Sistematika Laporan.....	4
1.7 Relevansi	4
BAB II TEORI DASAR	5
2.1 Tinjauan Pustaka	5
2.2 Pembudidayaan Ikan Gurami	7
2.3 Motor DC dan <i>Gearbox</i>	8
2.3.1 Sistem <i>Gearbox Worm Gear</i>	9
2.4 Arduino Mega	11
2.4.1 Pemrograman Arduino IDE.....	11
2.5 Sensor Cahaya Fotodioda.....	12
2.6 Potensiometer	12
2.7 DS3231 Sebagai <i>Real Time Clock</i>	13
2.8 Relay 13	13
2.9 LCD 20 x 4 dan Modul <i>Inter Integrated Circuit</i> LCD	14
2.10 Keypad 3x4	16
2.11 Rangkaian Dioda Pengaman	17

BAB III PERANCANGAN DAN PEMBUATAN ALAT	19
3.1 Penyusunan Perangkat Elektronik	20
3.1.1 Pembagian Pin Arduino	20
3.1.2 Pengkabelan <i>Driver Relay</i> Motor DC	20
3.1.3 Koneksi I2C Bus RTC DS3231 dan LCD	21
3.1.4 Perangkaian Sensor <i>Level</i> Fotodioda	22
3.1.5 Rangkaian Sensor Hujan	22
3.1.6 Perangkaian <i>Driver Relay</i> Lampu Indikator	23
3.1.7 Perangkaian <i>Driver Relay Blower</i>	23
3.1.8 Pengkabelan <i>Keypad</i> 3x4	24
3.2 Pembuatan Perangkat Mekanik	24
3.2.1 Pembuatan Tangki Pakan	25
3.2.2 Pembuatan Mekanisme Blower	27
3.3 Pembuatan Perangkat Lunak	29
3.3.1 Pembuatan <i>Flowchart</i> Program	30
3.3.2 Pendefinisian Pin dan Deklarasi Variabel	35
3.3.3 Bagian Program untuk Pengaturan Awal	36
3.3.4 Bagian Program untuk Membaca Posisi dan Mengatur Motor	37
3.3.5 Bagian Program Untuk Memberi Jeda Berhenti	37
3.3.6 Bagian Program Untuk Mengembalikan Posisi Sudut pada Titik Awal dan Berhenti	38
3.3.7 Bagian Program untuk Membuat Fungsi	39
3.3.8 Bagian Program Logika Indikator	39
BAB IV PENGUKURAN DAN PENGUJIAN	41
4.1 Data Pengujian Peralatan	41
4.1.1 Pengujian LCD 20x4	41
4.1.2 Pengujian <i>Keypad</i>	43
4.1.3 Pengujian Koneksi Sensor	44
4.1.4 Pengujian Motor DC	45
4.1.5 Pengujian <i>Real Time Clock</i>	45
4.1.6 Pengujian <i>Relay</i>	46
4.1.7 Pengujian Indikator Tangki	47
4.2 Pengukuran dan Pengujian Sensor <i>Level</i> Tangki	49
4.3 Pengukuran dan Pengujian Pergerakan Motor DC pada Buka-Tutup <i>Valve</i> Tangki Pakan	50
4.4 Pengukuran dan Pengujian Pergerakan Motor DC Pengatur Posisi <i>Blower</i>	52

4.5 Pengukuran dan Pengujian Keluaran Sensor Hujan	52
4.6 Pengujian Jarak Lontar Pakan dengan Blower	53
BAB V PENUTUP.....	55
5.1 Kesimpulan	55
5.2 Saran	56
DAFTAR PUSTAKA	57
LAMPIRAN.....	59

-----Halaman ini sengaja dikosongkan-----

DAFTAR GAMBAR

HALAMAN

Gambar 2.1 Model Mekanik Pemberi Pakan Ikan Jenis Lubang Geser Satu Lubang dan Dua Lubang.....	5
Gambar 2.2 Model Mekanik Pemberi Pakan Ikan Jenis Sentrifugal dan <i>Spindle</i>	6
Gambar 2.3 Model Mekanik Pemberi Pakan Ikan Jenis <i>Screw Conveyor</i> , <i>Belt Conveyor</i> dan <i>Screw Blower</i>	6
Gambar 2.4 Pakan Ikan yang digunakan jenis 781 -3	7
Gambar 2.5 Ikan Gurami Usia Dewasa.....	8
Gambar 2.6 Arah Medan Magnet Dalam Suatu Penghantar	9
Gambar 2.7 Interaksi Fluks Magnet Angker Dinamo dan Kutub Medan	9
Gambar 2.8 Motor DC <i>Gearbox</i> yang Digunakan	10
Gambar 2.9 Konfigurasi <i>Gearbox Worm Gear</i>	10
Gambar 2.10 Bentuk Fisik Arduino Mega 2560	11
Gambar 2.11 Jendela Arduino IDE Versi 1.6.8	11
Gambar 2.12 Simbol dari Fotodioda	12
Gambar 2.13 LCD <i>Display</i> Ukuran 20x4 yang Digunakan	14
Gambar 2.14 Modul I2C LCD yang Digunakan	16
Gambar 2.15 <i>Keypad</i> 3x4 yang Digunakan	17
Gambar 2.16 Pemasangan Dioda Pengaman pada Rangkaian yang Menggunakan Kumpanan.....	18
Gambar 3.1 Bagan Alur Kerja dari Alat	19
Gambar 3.2 Pengkabelan <i>Relay</i> 4 <i>Channel</i> dengan Arduino	21
Gambar 3.3 Pengkabelan I2C Bus dari Modul RTC DS3231 dan Modul I2C LCD Ukuran 20x4	21
Gambar 3.4 Rangkaian Sensor Pendeteksi <i>Level</i> dengan Fotodioda ke Pin Analog Arduino Dan LED Inframerah	22
Gambar 3.5 Rangkaian Modul Sensor Hujan ke Arduino dan Penghantar	23
Gambar 3.6 Rangkaian <i>Driver Relay</i> untuk Lampu Indikator	23
Gambar 3.7 Perangkaian <i>Driver Relay</i> untuk <i>Blower</i> Pakan	24
Gambar 3.8 Pengkabelan <i>Keypad</i> 3x4 ke Arduino Mega	24

Gambar 3.9 Desain Ukuran Tangki	25
Gambar 3.10 Desain <i>Valve Output</i> Pakan Ikan.....	26
Gambar 3.11 Bagian Depan Dan Belakang Rangka Penahan Tangki ...	26
Gambar 3.12 Bagian Samping Dan Atas Rangka Penahan Tangki	27
Gambar 3.13 Penempatan Sensor Fotodiode Dan LED Inframerah Pada Dinding Tangki Pakan	27
Gambar 3.14 Bagian Poros Penahan <i>Blower</i>	28
Gambar 3.15 Bagian Penahan Motor DC Tampak Dari Atas	29
Gambar 3.16 Bagian Penahan Motor DC Dari Depan Dan Samping	29
Gambar 3.17 <i>Flowchart</i> Program untuk Menjatuhkan Pakan Melalui Valve Tangki	31
Gambar 3.18 <i>Flowchart</i> Program Indikator Tangki	32
Gambar 3.19 <i>Flowchart</i> Program untuk Mengatur Waktu Pakan Melalui Antarmuka LCD dan <i>Keypad</i>	33
Gambar 3.20 <i>Flowchart</i> Program Pemberian Pakan	34
Gambar 3.21 Bagian Pendefinisian Pin dan Deklarasi Variabel.....	35
Gambar 3.22 Pendefinisian <i>Keypad</i> dan LCD <i>Display</i>	35
Gambar 3.23 Bagian Pengaturan Awal Program	36
Gambar 3.24 Bagian Program Pembacaan Posisi dan <i>Starting Motor</i> ..	37
Gambar 3.25 Bagian Program untuk Memberi Jeda dengan <i>Millis()</i>	37
Gambar 3.26 Bagian Program untuk Mengembalikan Posisi Motor dan Berhenti.....	38
Gambar 3.27 Bagian Program untuk Membuat Fungsi	39
Gambar 3.28 Bagian Program Logika untuk Indikator.....	39
Gambar 4.1 Gambar Keseluruhan Alat Disertai Kolam dan <i>Box</i> Kontroler	41
Gambar 4.2 Program pengujian fungsi LCD	42
Gambar 4.3 Tampilan Layar LCD Dari Pengujian Program	42
Gambar 4.4 Program untuk Menguji Karakter LCD	42
Gambar 4.5 Tampilan Layar LCD Hasil Pengujian Karakter	43
Gambar 4.6 Program Pengujian <i>Keypad</i>	43
Gambar 4.7 Tampilan pada <i>Serial Monitor</i> Saat Masing-Masing Tombol Ditekan.....	44
Gambar 4.8 Pembacaan Nilai Sensor pada <i>Serial Monitor</i>	45
Gambar 4.9 Program yang Digunakan untuk Pengujian RTC	46
Gambar 4.10 Tampilan Waktu Hasil Pengujian RTC pada LCD <i>Display</i>	46
Gambar 4.11 Waktu pada Jendela <i>Date and Time Windows</i>	46
Gambar 4.12 Kondisi Ketiga Lampu Menyala	48

Gambar 4.13 Kondisi Satu Lampu Mati	48
Gambar 4.14 Kondisi Dua Lampu Mati	48
Gambar 4.15 Kondisi Dua Lampu Mati dan Satu Lampu Berkedip	49
Gambar 4.16 Pembacaan ADC Fotodioda Pada Saat Terdapat Pakan...	50
Gambar 4.17 Pembacaan ADC Fotodioda Pada Saat Tidak Terhalang ..	50
Gambar 7.1 Fisik Alat Pemberi Pakan dari Samping.....	83
Gambar 7.2 Fisik Alat Pemberi Pakan dari Belakang	83
Gambar 7.3 Fisik Alat Pemberi Pakan dari Depan	84
Gambar 7.4 Pemasangan Blower pada Motor Pengatur Sudut	84
Gambar 7.5 <i>Box</i> Kontrol dengan Antarmuka LCD dan Keypad.....	85
Gambar 7.6 Rangkaian Dalam <i>Box</i> Kontrol.....	85
Gambar 7.7 Kolam untuk Simulasi Pemberian Pakan dan Pengaturan Sirkulasi Air	86
Gambar 7.8 Pemasangan Modul Sensor Hujan	86

-----Halaman ini sengaja dikosongkan-----

DAFTAR TABEL

HALAMAN

Tabel 2.1 Tabel Fungsi – Fungsi Pin pada LCD <i>Display</i> 20x4	15
Tabel 2.2 Tabel Fungsi Tiap Pin pada Modul I2C LCD	16
Tabel 3.1 Pin Mapping I/O Arduino Mega	20
Tabel 4.1 Hasil Pengujian Sensor – Sensor yang Ada	44
Tabel 4.2 Tabel Pengujian Motor DC	45
Tabel 4.3 Tabel Kondisi <i>Relay</i> Saat Diuji	47
Tabel 4.4 Tabel Kondisi Tiap Lampu Saat Diuji	47
Tabel 4.5 Hasil Pengujian Program Indikator Tangki	48
Tabel 4.6 Pengukuran Sensor Fotodioda	49
Tabel 4.7 Data Pengukuran Sensor Posisi Potensiometer 1	51
Tabel 4.8 Data Pengukuran Sensor Posisi Potensiometer 2	52

-----Halaman ini sengaja dikosongkan-----

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pemberian pakan ikan adalah salah satu hal penting dalam pembudidayaan ikan. Sayangnya pada saat ini sistem pemberian pakan ikan umumnya masih sangat bergantung pada sumber daya manusia dan untuk pemberiannya dilakukan secara manual. Pemberian pakan dilakukan secara sederhana yaitu menyebar pakan ikan dengan tangan langsung ke arah kolam ikan. Sehingga hal ini akan menyebabkan lamanya pemberian pakan pada ikan bila seorang petani tersebut mempunyai lahan kolam yang banyak. Apalagi jika seorang petani ikan tersebut lupa atau terlambat dalam memberi pakan ikan, maka juga akan menyebabkan tidak teraturnya jadwal pemberian pakan ikan. Metode pemberian pakan yang terbaik yaitu dengan menabur pakan dengan rata dan tersebar diseluruh permukaan kolam. Jika pemberian pakan tidak teratur, maka akan berdampak pada pertumbuhan ikan yang menjadi kurang maksimal dan disaat panen nantinya ukuran ikan menjadi tidak setara antara satu dengan yang lain.

Pada keadaan tertentu misalnya disaat kondisi hujan, ikan tidak diberi pakan. Hal tersebut dikarenakan ikan gurami tidak nafsu makan ketika cuaca sedang hujan. Akibatnya akan sia-sia jika tetap dilakukan pemberian pakan.

Sebenarnya sudah ada alat pemberi pakan ikan otomatis yang beredar, namun dari segi penyebaran masih terpusat disuatu titik tertentu dan tidak merata. Hal tersebut menyebabkan alat tersebut bekerja kurang efektif dan masih terdapat kekurangan. Karena jika petani ikan ingin mendapatkan hasil panen yang maksimal, maka syarat utama yaitu pada pemberian pakannya yang harus merata sehingga pertumbuhan antar ikan menjadi setara.

Diharapkan dengan terselesaikannya alat ini, tingkat efisiensi peternakan ikan gurami bisa semakin baik dan dapat meningkatkan taraf hidup maupun kesejahteraan petani ikan gurami.

1.2 Permasalahan

Pemberian pakan ikan masih bersifat manual. Selain itu, jika peternak sedang berada diluar rumah atau bepergian jauh maka tidak ada lagi yang memberi pakan ikannya. Hal tersebut akan berpengaruh ke

faktor pertumbuhan dan kesehatan ikan. Apabila menggunakan tenaga kerja manusia maka akan menambah ongkos budidaya untuk menggaji karyawan.

1.3 Batasan Masalah

Batasan Masalah dalam tugas akhir ini antara lain yaitu :

1. Ikan yang menjadi objek budidaya adalah ikan gurami konsumsi.
2. Tahapan budidaya gurami yang menjadi obyek adalah tahap pembesaran.
3. Pakan ikan yang digunakan berupa *pellet* pabrikan standar jenis 781-3 yang umum digunakan untuk masa akhir budidaya.
4. Kontroler yang digunakan adalah Arduino Mega 2560.
5. Motor DC yang digunakan adalah motor DC yang dilengkapi *Gearbox* dengan konfigurasi *worm gear*.
6. Pengaturan posisi motor DC menggunakan potensiometer rotari.
7. *Real Time Clock* menggunakan IC DS3231.

1.4 Tujuan

Tujuan kami menuliskan tugas akhir ini adalah:

- Mengotomatisasi sistem pemberian pakan ikan yang semula dilakukan secara manual dengan mesin pemberian pakan yang bekerja secara terjadwal dan bisa merata saat menebar pakan.
- Menciptakan sebuah alat yang dapat mempermudah dan tepat guna untuk digunakan para petani ikan supaya bisa meningkatkan hasil panen maupun kesejahteraan hidup mereka.

1.5 Metodologi Penelitian

Penelitian ini dilakukan melalui beberapa tahapan metodologi, yaitu, tahap persiapan, tahap perencanaan dan pembuatan alat, tahap pengujian dan analisis, dan yang terakhir adalah penyusunan laporan berupa buku Tugas Akhir.

1. Tahap Persiapan

Pada tahap ini dipelajari sumber-sumber dan literatur yang berhubungan dengan pengerjaan alat, antara lain :

- a. Mempelajari pakan yang digunakan untuk ikan gurami

- b. Mempelajari metode yang sesuai untuk memberi pakan pada ikan gurami.
- c. Mempelajari penggunaan potensiometer untuk membatasi gerak motor DC.
- d. Mempelajari penggunaan *relay* untuk mengaktifkan dan mengubah polaritas motor DC.
- e. Mempelajari literatur yang berhubungan dengan pembudidayaan ikan gurami.

2. Tahap Perencanaan dan Pembuatan Alat

Pada tahap ini mulai dirancang rangkaian elektrik dan mekanis dari alat serta membuat program awal dari keseluruhan rangkaian.

Proses yang dilalui antara lain yaitu :

- a. Merancang konstruksi mekanis dari alat meliputi mekanika tangki dan mekanika *blower*.
- b. Merancang rangkaian daya dan kontrol elektrik untuk mengendalikan keseluruhan alat.
- c. Membangun konstruksi fisik dari mekanis alat hingga menjadi satu kesatuan.
- d. Merangkai rangkaian elektrik dari alat dengan menyesuaikan peralatan mekanis yang digunakan.
- e. Menyusun program kendali alat yang dibutuhkan berdasarkan rangkaian listrik yang ada.

3. Tahap Pengujian dan Analisis Data

Pada tahap ini rangkaian yang sudah jadi diujicoba apakah dapat berjalan sesuai keinginan dan apakah data yang didapat bagus atau tidak. Proses yang dilalui antara lain yaitu :

- a. Menguji apakah rangkaian mekanis dapat bekerja sesuai dengan rancangan awal.
- b. Memperbaiki konstruksi mekanis yang tidak cocok untuk direalisasikan ataupun tidak sesuai.
- c. Menguji apakah komponen-komponen yang ada pada rangkaian elektrik dapat bekerja sebagaimana mestinya.
- d. Menguji apakah tiap sub bagian dari rangkaian dapat bekerja dengan baik.
- e. Menguji apakah program dapat menjalankan rangkaian dengan baik dan memperbaiki susunan program yang tidak sesuai.
- f. Mengambil data dari kinerja alat untuk dilihat seberapa baik alat dapat bekerja.

4. Tahap Penyusunan Buku Laporan

Setelah alat dapat berjalan dengan baik disusun buku laporan berdasarkan riwayat pengerjaan dan data yang diambil dari alat. Diharapkan buku yang disusun dapat bermanfaat bagi semua orang yang ingin mempelajari pembuatan alat pemberi makan ikan otomatis khususnya dengan metode yang kami rancang.

1.6 Sistematika Laporan

Pembahasan Tugas Akhir ini akan dibagi menjadi lima Bab dengan sistematika sebagai berikut:

Bab I Pendahuluan

Bab ini meliputi latar belakang, permasalahan, tujuan penelitian, metodologi penelitian, sistematika laporan, dan relevansi.

Bab II Teori Dasar

Bab ini menjelaskan tentang tinjauan pustaka, konsep dari Arduino, Motor DC *Gearbox*, *Relay*, *Real Time Clock* dan Sensor Fotodioda.

Bab III Perancangan dan Pembuatan Alat

Bab ini membahas tentang cara pembuatan alat baik secara mekanis, elektrik, dan perangkat lunaknya.

Bab IV Pengujian dan Pengambilan Data

Bab ini memuat tentang pemaparan dan analisis hasil pengujian alat pada keadaan sebenarnya. Seperti pengujian aktivitas motor DC, *power supply* dan sensor pendeteksi *level*. Pada tiap pengujian akan ada analisis terkait metode yang digunakan.

Bab V Penutup

Bab ini berisi kesimpulan dan saran dari hasil pengambilan data yang telah diperoleh.

1.7 Relevansi

Hasil yang diperoleh dari Tugas Akhir ini diharapkan memberikan manfaat berupa penerapannya pada budidaya ikan gurami skala luas sehingga diharap proses budidaya semakin meningkat kualitasnya dan dapat meningkatkan ketahanan pangan nasional.

BAB II

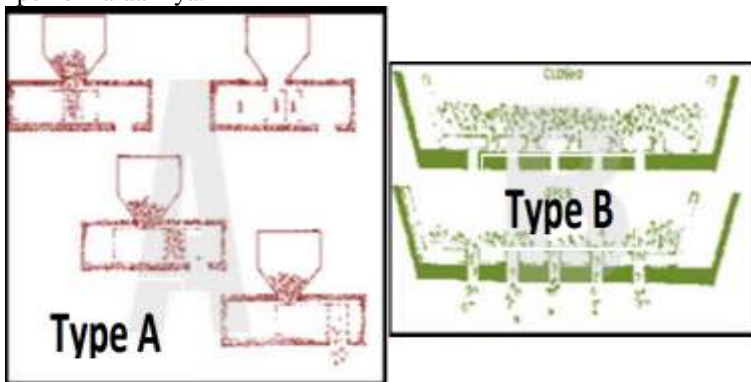
TEORI DASAR

Bab ini akan dibahas mengenai teori – teori yang berkaitan dengan peralatan yang akan dirancang. Teori yang mendukung penyelesaian tugas akhir ini diantaranya adalah mengenai; motor DC *Gearbox*, Arduino Mega, Lampu LED, Fotodiode, *Power Supply* 24 Volt DC.

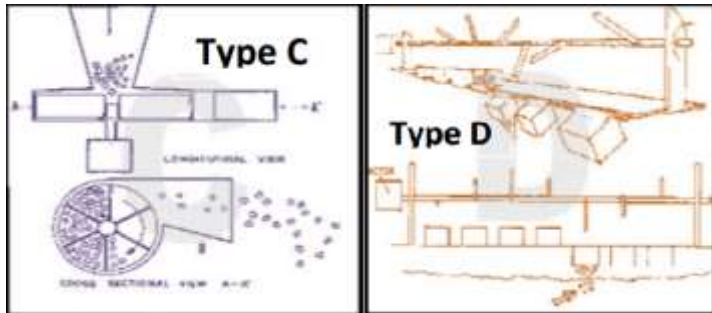
2.1 Tinjauan Pustaka

Dalam perancangan alat pemberi makan ikan gurami otomatis ini, digunakan Arduino Mega sebagai kontroler utama bagi sistem dan sebuah *blower* sebagai pelontar pakan. Untuk memenuhi spesifikasi alat pemberi makan ikan yang sesuai dengan harapan, digunakan beberapa komponen tambahan seperti motor DC *Gearbox*. Kemudian untuk antarmuka dengan pengguna digunakan peralatan *input* dan *output* seperti *keypad* dan LCD *display* 20x4.

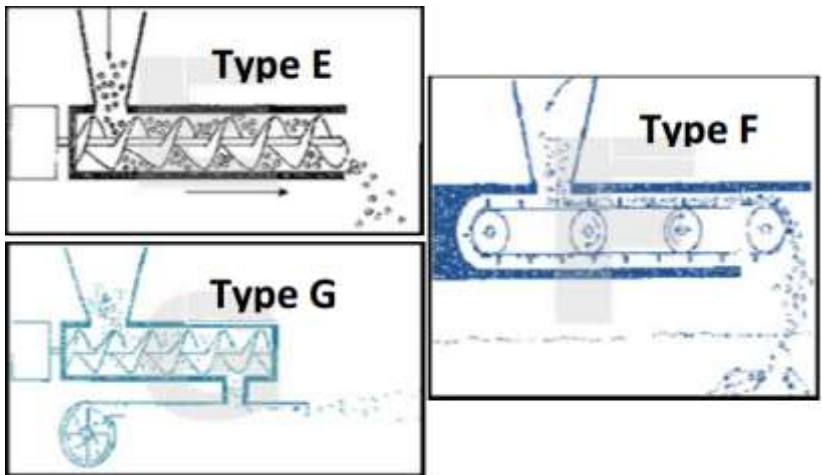
Dalam pembudidayaan ikan yang terotomasi terutama di negara negara maju, sudah ada beberapa model mekanis untuk memberikan pakan secara otomatis seperti menggunakan *screw conveyor*, *valve* elektromagnetik, pelontar sentrifugal, *feed hopper*, dan lain sebagainya. Namun masing masing model mekanik diatas belum tentu cocok untuk diterapkan pada ikan gurami, dikarenakan ikan gurami termasuk sensitif dalam pemeliharaannya.



Gambar 2.1 Model Mekanik Pemberi Pakan Ikan Jenis Lubang Geser Satu Lubang dan Dua Lubang[7]



Gambar 2.2 Model Mekanik Pemberi Pakan Ikan Jenis Sentrifugal dan *Spindle*[7]



Gambar 2.3 Model Mekanik Pemberi Pakan Ikan Jenis *Screw Conveyor*, *Belt Conveyor* dan *Screw* Dilengkapi *Blower*[7]

Salah satu metode yang ada adalah menggunakan *blower* untuk mendorong pakan ikan terlontar lebih jauh. Akan tetapi pada penerapan untuk ikan gurami diperlukan modifikasi khusus yaitu pengaturan sudut lempar untuk mengatur jauh dekatnya lontaran pakan.

Feeder otomatis ini dirancang untuk dapat dengan mudah dioperasikan oleh orang awam sekalipun, oleh karena itu antarmukanya dibuat sederhana. Pengguna cukup mengatur waktu pemberian pakan dan kontroler akan mengaktifkan sistem *feeder* ketika waktu sistem dari RTC bersesuaian dengan waktu yang ditentukan. Saat sistem aktif, motor DC akan menggerakkan moncong *blower* ke posisi nol derajat. Kemudian

blower aktif dan motor yang terdapat di tangki pakan akan menggerakkan *valve* untuk menjatuhkan pakan ke arah *blower* sehingga akan terlontar ke arah kolam ikan. Sesaat kemudian *valve* akan tertutup dan motor DC akan merubah derajat posisi *blower* sebesar 15 derajat keatas. Begitu seterusnya sampai *blower* mencapai sudut 45 derajat, kemudian posisi sudut *blower* akan dikembalikan ke posisi nol derajat.

Penentuan derajat posisi *blower* didasarkan pada pengaturan persebaran pakan yang diambil dari pengujian langsung ketika *blower* digunakan secara langsung untuk meniup pelet pakan ikan. Sehingga dengan perubahan derajat peniupan pakan tersebut persebaran pakan dapat merata dan pertumbuhan ikan dalam satu kolam juga merata.



Gambar 2.4 Pakan Ikan yang digunakan jenis 781 -3

Untuk pakan ikan yang digunakan distandarkan untuk jenis ikan gurami berusia 7 bulan keatas (dewasa/menjelang masa panen), dan contoh pakan ikan yang kami gunakan adalah tipe 781 – 3 produksi PT. Charoen Pokphand Prima. Pada usia tersebut, ikan sedang pesat tumbuh sehingga pakan dengan protein tinggi sangat cocok untuk pembesaran ikan. Selain itu volume pakan yang harus ditebar juga semakin banyak dan berat seiring pertumbuhan ikan, oleh karena itu alat cocok diaplikasikan dikondisi ini.

2.2 Pembudidayaan Ikan Gurami

Ikan gurami merupakan salah satu komoditi perikanan air tawar yang cukup penting apabila dilihat dari permintaannya yang cukup besar dan harganya yang relatif tinggi dibandingkan dengan ikan air tawar lainnya seperti ikan mas, nila, tambakan, dan tawes. Ikan gurami merupakan salah satu sumber protein yang cukup tinggi. Bagi masyarakat umum, ikan ini dipandang sebagai salah satu ikan bergengsi dan biasanya disajikan pada

acara acara yang dianggap penting. Oleh sebab itu tidak mengherankan apabila ikan gurami menjadi salah satu komoditi unggulan di sektor perikanan air tawar. Ikan gurami relatif merupakan jens ikan pemakan tumbuhan atau bersifat herbivora, tetapi setelah dibudidayakan bersifat omnivora karena sudah diberi pakan berupa *pellet*. Masa pemeliharaannya relatif lama sehingga dilakukan dalam beberapa tahap pemeliharaan yaitu tahap pembenihan, tahap pendederan dan tahap pembesaran, dimana pada masing masing tahapan menghasilkan produk yang dapat dipasarkan secara tersendiri.



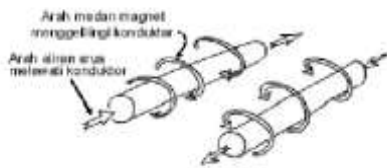
Gambar 2.5 Ikan Gurami Usia Dewasa

Pembesaran merupakan tahapan akhir dari pemeliharaan yang menghasilkan gurami siap konsumsi. Faktor yang perlu diperhatikan adalah ukuran kolam dengan bibit yang ditebar, kualitas air kolam/ lingkungan, pakan tambahan dan teknis budidayanya menggunakan monokultur atau polikultur. Ikan yang dipelihara dapat berukuran berat 200 – 250 gram/ekor dan ditebar dengan kepadatan benih kurang lebih 1- 2 kg per meter persegi. Untuk memudahkan pengelolaan pemberian pakan sebaiknya menggunakan takaran yang baku yaitu *pellet* diberikan sebanyak 2% biomasa/hari pada sore hari. Dalam waktu 4 bulan ikan akan mencapai ukuran konsumsi dengan berat 500 – 700 gram/ekor. Panen gurami tergantung permintaan konsumen.[6]

2.3 Motor DC dan Gearbox

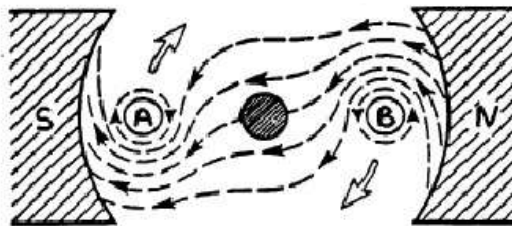
Motor listrik merupakan suatu alat yang mengubah energi listrik menjadi energi mekanik berupa putaran. Motor DC (*Direct Current*) sesuai namanya adalah motor listrik yang bekerja ketika mendapat arus searah.

Ketika arus mengalir melewati suatu penghantar, maka disekitar konduktor tersebut akan timbul medan magnet. Arah fluks medan magnet mengikuti arah arus sesuai dengan aturan tangan kanan.



Gambar 2.6 Arah Medan Magnet dalam Suatu Penghantar

Gambar 2.7 menunjukkan medan magnet yang terbentuk disekitar penghantar berbentuk U. Pada motor listrik penghantar berbentuk U seperti ini disebut dengan angker dinamo. Jika angker dinamo diletakkan diantara kutub utara dan selatan yang cukup kuat, maka medan magnet penghantar dan medan magnet kutub akan berinteraksi.



Gambar 2.7 Interaksi Fluks Magnet Angker Dinamo dan Kutub Medan

Pada gambar diatas, arus masuk dari arah A dan keluar dari arah penghantar B. Akibat interaksi medan, medan yang searah akan saling menguatkan, sedangkan medan yang berlawanan akan saling melemahkan. Akibat adanya medan yang kuat, penghantar akan berusaha keluar dari medan tersebut dan terdorong ke wilayah yang medannya lemah. Proses ini terjadi secara berkesinambungan selama ada arus yang mengalir sehingga angker motor berputar searah jarum jam.[9]

2.3.1 Sistem *Gearbox Worm Gear*

Dalam suatu konstruksi mekanis, terutama motor listrik seringkali kita dihadapkan pada permasalahan dengan kemampuan mekaniknya yang tidak sesuai dengan spesifikasi yang dibutuhkan. Ada motor listrik yang berputar dengan kecepatan tinggi namun kekuatan putar/torsinya kecil. Sebaliknya

ada pula motor listrik yang berputar pelan dengan torsi yang besar. Sistem *gearbox* merupakan solusi mekanis untuk memanipulasi kecepatan dan torsi suatu motor DC. Selain itu *gearbox* juga dapat merubah karakteristik mekanis dari perputaran motor. Roda gigi merupakan salah satu elemen transmisi gerakan yang penting sebagai sarana perpindahan gaya.



Gambar 2.8 Motor DC *Gearbox* yang Digunakan

Worm gear merupakan salah satu konfigurasi roda gigi yang umum digunakan sebagai pengunci gerakan. Ini dikarenakan sesuai konstruksinya batang cacing dapat menggerakkan roda gigi dengan perputaran ulirnya. Namun sebaliknya roda gigi tidak bisa menggerakkan batang cacing secara langsung karena tertahan oleh ulir cacing. Sehingga perputaran roda gigi hanya dapat terjadi bila batang cacing yang digerakkan.



Gambar 2.9 Konfigurasi *Gearbox Worm Gear*

2.4 Arduino Mega

Arduino merupakan rangkaian mikrokontroler dalam satu papan yang bersifat *open source*. Arduino adalah kombinasi dari perangkat keras, bahasa pemrograman dan *Integrated Development Environment (IDE)* yang canggih.



Gambar 2.10 Bentuk Fisik Arduino Mega 2560

Arduino Mega merupakan sebuah board mikrokontroler yang berbasis Atmega 2560. Arduino Mega memiliki 54 pin input/output (14 diantaranya dapat digunakan untuk fungsi PWM), 16 Analog Input, 4 UART, Osilator Kristal 16 MHz, koneksi USB, serta tombol *reset*.

2.4.1 Pemrograman Arduino IDE

IDE adalah sebuah aplikasi untuk menulis program, meng-*compile* menjadi kode biner dan meng-*upload* ke dalam memori mikrokontroler. Ada banyak proyek dan alat-alat yang dikembangkan oleh akademisi dan profesional dengan menggunakan Arduino. Selain itu juga ada banyak modul-modul pendukung (sensor, tampilan, penggerak dan sebagainya) yang dibuat oleh pihak lain untuk dapat terhubung dengan Arduino.



Gambar 2.11 Jendela Arduino IDE Versi 1.6.8

2.5 Sensor Cahaya Fotodioda

Fotodioda merupakan sensor yang sensitif terhadap cahaya yang mengenainya. Komponen semikonduktor ini dapat mengubah cahaya menjadi arus listrik. Sensitivitas fotodioda bergantung pada jenis spektrum cahaya yang mengenainya seperti sinar tampak, sinar inframerah dan sinar *ultraviolet*. Aplikasi sensor fotodioda pada alat kami yaitu digunakan untuk mendeteksi *level* pakan yang ada didalam tangki, sehingga ketika sensor terhalang oleh pakan yang ada didalam maka tegangan analog yang dideteksi ikut berubah, begitu pula sebaliknya. Bentuk fisik fotodioda sangat mirip dengan *LED (Light Emitting Diode)* hanya saja tidak memancarkan sinar ketika dicatu dengan sumber tegangan.[8]



Gambar 2.12 Simbol dari Fotodioda

2.6 Potensiometer

Potensiometer merupakan resistor yang resistansinya dapat berubah ubah. Berdasarkan bentuknya potensiometer ada 3 jenis yaitu potensiometer *slider*, rotari, dan *trimmer*. Potensiometer rotari merupakan jenis potensiometer yang resistansinya berubah ketika asnya diputar.

Dua pin pada potensiometer membentuk sebuah jalur elemen resistif dengan *wiper* sebagai penentu rasio resistansinya. Dengan sifat perubahan resistansi tersebut potensiometer seringkali digunakan sebagai *input* maupun sensor.



Gambar 2.8 Ilustrasi Potensiometer

2.7 DS3231 Sebagai *Real Time Clock*

Real Time Clock merupakan rangkaian yang berfungsi memberikan sinyal waktu seperti layaknya sebuah jam. Selain dapat menyimpan data dalam format jam, menit, dan detik, *Real Time Clock* pada umumnya dapat menyimpan tanggal dalam format tanggal, bulan, tahun serta 7 hari dalam seminggu.

DS3231 merupakan salah satu IC *Real Time Clock* keluaran Maxim yang memiliki akurasi tinggi. DS3231 juga mendukung fitur I2C untuk mempermudah komunikasi serial dengan kontroler. IC ini memiliki kristal osilator internal didalam chip-nya disertai dengan sensor suhu untuk mengatasi perubahan frekuensi kristal saat temperatur lingkungannya berubah ubah. Karena dilengkapi dengan baterai backup 3,3 V, pengaturan waktu akan tetap berjalan walaupun sumber daya utama dimatikan sehingga meski terjadi pemutusan daya terhadap sistem akibat mati listrik dan sebagainya, waktu tetap akurat. Bentuk fisik modul RTC DS3231 dapat diamati pada Gambar 2.9 berikut :



Gambar 2.9 Modul RTC DS3231

2.8 *Relay*

Relay merupakan komponen yang berfungsi memutus dan menyambung rangkaian dengan memanfaatkan suatu kumparan. *Relay* memiliki 2 jenis kontak yaitu *Normally Open* (NO) dan *Normally Close* (NC), kontak NO posisi awalnya sebelum aktif adalah *open* atau terputus sedangkan kontak NC posisi awalnya *close* atau tersambung. Posisi kedua kontak akan berubah bersamaan ketika *relay* aktif, yaitu saat kumparannya mendapat tegangan input dari kontroler. Modul *relay* yang berada di pasaran pada umumnya telah dilengkapi dengan pengaman berupa dioda untuk mencegah arus balik atau kesalahan penyambungan dan lampu indikator. Pada alat kami, digunakan 2 jenis *driver relay*, yaitu modul *driver*

relay 1 channel jenis *active high* dimana kontak *relay* akan aktif saat menerima sinyal *HIGH*, dan modul *driver relay 4 channel* jenis *active low* seperti pada Gambar 2.10, dimana kontak *relay* akan aktif saat menerima sinyal *LOW* dari arduino.



Gambar 2.10 Modul Relay 4 Channel

2.9 LCD 20 x 4 dan Modul Inter Integrated Circuit LCD

Liquid Crystal Display atau biasa disebut LCD digunakan untuk menampilkan karakter huruf angka maupun simbol melalui program di mikrokontroler. Sesuai dengan ukurannya, LCD 20x4 memiliki 20 kolom dan 4 baris yang dapat diisi karakter. Dalam alat kami, LCD *display* digunakan sebagai antarmuka bagi pengguna untuk menjalankan alat serta mengetahui kondisi alat. Pengaturan jam pakan dan waktu *real time* saat ini ditampilkan melalui tampilan LCD *display*.



Gambar 2.13 LCD Display Ukuran 20x4 yang Digunakan

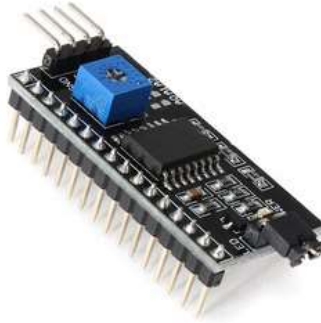
Sedangkan fungsi dari setiap pin LCD adalah sebagai berikut :

Tabel 2.1 Tabel Fungsi – Fungsi Pin pada LCD *Display* 20x4 [4]

No	Nama Pin	Fungsi
1	Vss	<i>Ground</i>
2	Vdd	Suplai tegangan +5V
3	Vlc	Tegangan kontras LCD
4	RS	L = <i>Input</i> instruksi H = <i>Input</i> data
5	R/W	L = Tulis data dari MPU ke LCM H = Baca data dari LCM ke MPU
6	<i>E</i>	<i>Enable clock</i>
7	DB0	<i>Data bus line 0</i>
8	DB1	<i>Data bus line 1</i>
9	DB2	<i>Data bus line 2</i>
10	DB3	<i>Data bus line 3</i>
11	DB4	<i>Data bus line 4</i>
12	DB5	<i>Data bus line 5</i>
13	DB6	<i>Data bus line 6</i>
14	DB7	<i>Data bus line 7</i>
15	Anoda	Tegangan positif <i>backlight</i>
16	Katoda	Tegangan negatif <i>backlight</i>

LCD memiliki peran penting dalam sistem alat kami, terutama untuk menampilkan status kolam ikan dan antarmuka yang dapat diakses oleh pengguna alat.

Untuk memangkas jumlah pin yang digunakan pada arduino sehingga lebih ringkas dan efisien, maka digunakan modul *Inter Integrated Circuit* (I2C) untuk LCD sehingga penggunaan 16 pin menuju mikrokontroler dapat dipangkas menjadi 4 pin saja. Modul ini sudah didesain dengan *pinout* sedemikian rupa sehingga mudah dipasang pada pin LCD.



Gambar 2.14 Modul I2C LCD yang Digunakan

Pin yang terdapat didalam modul I2C LCD ini diantaranya adalah :

Tabel 2.2 Tabel Fungsi Tiap Pin pada Modul I2C LCD

No	Nama Pin	Fungsi
1	Gnd	<i>Ground</i>
2	Vcc	Sumber tegangan +5V
3	SDA	<i>Serial Data</i>
4	SCL	<i>Serial Clock</i>

Komunikasi *Inter Integrated Circuit* (I2C) merupakan konsep dasar komunikasi 2 arah antar IC maupun antar sistem secara serial menggunakan 2 kabel (*Two Wire Protocol*). Keduanya adalah *Serial Data* (SDA) dan *Serial Clock* (SCK). Setiap IC yang terhubung dengan I2C memiliki alamat khusus yang dapat diakses melalui *software*.

Data *bit* dikirim/diterima melalui SDA, sedangkan sinyal *clock* dikirim/diterima melalui SCL. Dalam tiap transfer data *bit* dihasilkan satu sinyal *clock*. Transfer data *bit* dianggap valid apabila data bit pada SDA tetap stabil selama *clock* ber kondisi *HIGH*. Data *bit* hanya boleh berubah ketika *clock* ber kondisi *LOW*.

2.10 Keypad 3x4

Keypad merupakan peralatan input utama untuk menjalankan alat kami. Pengaturan waktu pakan dan pemilihan menu pada antarmuka yang ditampilkan melalui LCD memanfaatkan pembacaan tombol *keypad* yang ditekan. Umumnya *keypad* digolongkan berdasarkan jumlah tombolnya dan disebut berdasarkan jumlah kolom dan barisnya. Jenis *keypad* yang umum dijual di pasaran antara lain yaitu *keypad* 3x3, 3x4, serta 4x4.



Gambar 2.15 Keypad 3x4 yang Digunakan

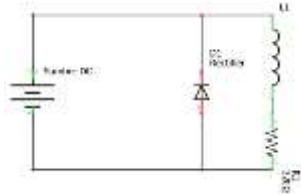
Pada dasarnya *keypad* hanya berisi susunan saklar-saklar biasa, yang sering digunakan adalah saklar *push button*. Cara kerjanya hanya berdasarkan proses konduktansi pada saat tombol ditekan. Tombol yang tertekan oleh jari, akan mendorong lempeng konduktor yang melekat pada lapisan pegas menyentuh lapisan konduktor yang terdapat di bagian dasar (lapisan atas dari PCB). Selanjutnya hal ini akan mengakibatkan terminal-terminal konduktor yang ada di dasar saling berhubungan. Jika saja pada kedua terminal tersebut diberi beda potensial maka pada saat tombol dilepas, tidak akan ada arus listrik yang mengalir diantara kedua terminal. Sebaliknya, saat tombol ditekan, maka konduktor yang terdapat pada bagian dasar pegas akan menyentuh kedua terminal tersebut. Dengan kata lain akan ada arus listrik yang mengalir diantara kedua terminal tersebut.[2]

Dengan demikian, tombol yang ditekan dapat dideteksi oleh mikrokontroler berdasarkan baris dan kolom keberapa yang mengalami penekanan dan dialiri arus.

2.11 Rangkaian Dioda Pengaman

Pada setiap peralatan listrik yang menggunakan kumparan seperti induktor, motor, dan *solenoid* akan timbul medan magnet saat kumparan tersebut dialiri arus listrik. Ketika arus listrik dihilangkan dari kumparan tersebut, medan magnet akan berangsur-angsur hilang dan menghasilkan arus listrik yang arahnya berlawanan dari arus yang mengalir sebelumnya, yaitu dari negatif menuju positif. Akibatnya hal ini berpotensi merusak komponen-komponen listrik yang dilewatinya.[1]

Dioda merupakan komponen elektronika yang memiliki sifat menghantarkan arus yang searah polaritasnya dan menahan arus yang polaritasnya berlawanan. Hal ini dapat dimanfaatkan untuk mencegah arus balik dari komponen magnetis seperti motor dengan memasangnya secara paralel pada rangkaian motor.



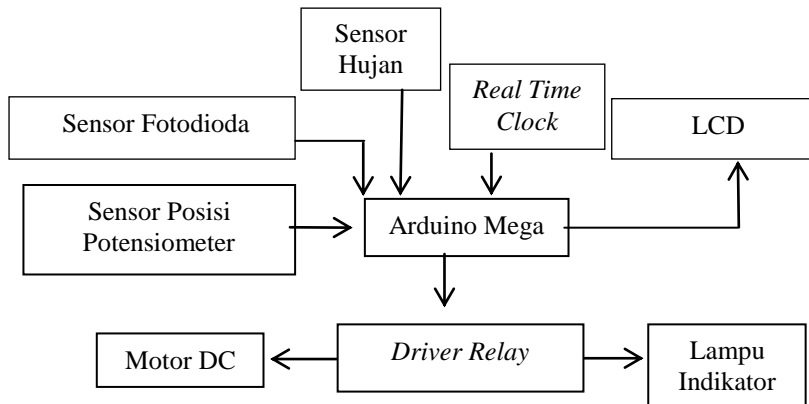
Gambar 2.16 Pemasangan Dioda Pengaman pada Rangkaian yang Menggunakan Kumputan

Dioda memiliki beragam jenis, seperti dioda penyearah (*rectifier*) dioda zener, dioda varactor, LED (*Light Emitting Diode*) serta fotodioda. Namun yang paling sering digunakan adalah dioda penyearah, umumnya digunakan pada proses penyearahan tegangan AC menjadi tegangan DC pada rangkaian *power supply*.

BAB III

PERANCANGAN DAN PEMBUATAN ALAT

Pada bab ini akan dijelaskan tahapan tahapan yang ada didalam perancangan dan pembuatan tugas akhir ini. Tahapan – tahapan yang dimaksud diantaranya adalah bagan alur kerja dari komponen komponen didalam alat, penyusunan perangkat – perangkat elektronik, pembuatan perangkat mekanik, serta pembuatan perangkat lunak bagi sistem.



Gambar 3.1 Bagan Alur Kerja dari Alat

Untuk mendapatkan alat pemberi makan ikan yang berjalan sesuai harapan diperlukan bagian pokok, yaitu:

1. Sensor Potensiometer, merupakan sensor yang mendeteksi posisi derajat motor DC berdasarkan perubahan nilai resistansinya.
2. Arduino Mega, mikrokontroler yang berfungsi sebagai pengendali, pengolah sinyal masuk dan keluar. Pin yang digunakan adalah pin *digital input*, *digital output*, dan *analog digital converter* sebagai *input analog*
3. *Driver Relay*, digunakan untuk mengubah polaritas motor DC sehingga dapat merubah arah putarnya.
4. Motor *DC Gearbox*, berfungsi sebagai penggerak *valve* pembuka tangki dan pengatur posisi *blower*.

Diagram blok pada Gambar 3.1 diatas menjelaskan mengenai alur kerja komponen – komponen *input/output* terhadap kontroler. Penjelasan

dari masing-masing komponen dijelaskan pada sub bab pembuatan perangkat elektronik, pembuatan perangkat mekanik, dan pembuatan perangkat lunak.

3.1 Penyusunan Perangkat Elektronik

Untuk menjalankan sistem *feeder* agar dapat berjalan dengan baik, diperlukan penyusunan yang tepat untuk komponen komponen elektronik yang ada.

3.1.1 Pembagian Pin Arduino

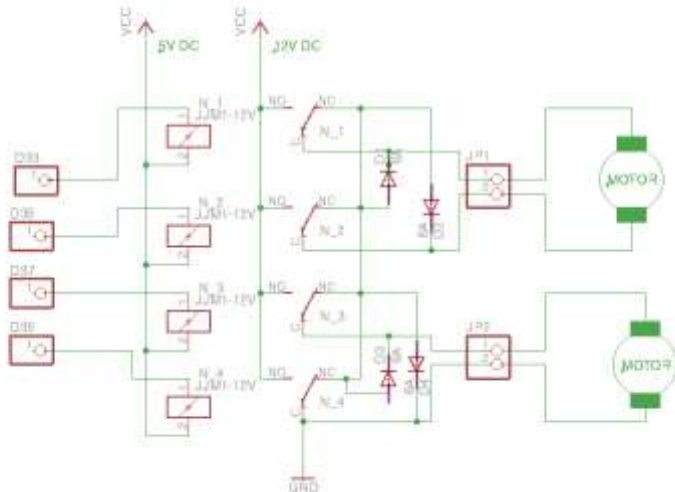
Seluruh komponen kendali baik input dan output akan diproses oleh arduino sebagai kontroler utama dari keseluruhan rangkaian. Pembagian pin yang I/O arduino adalah sebagai berikut :

Tabel 3.1 Pin Mapping I/O Arduino Mega

No	Nama Perangkat	Pin yang Digunakan
1	Sensor Hujan	A0
2	RTC DS3231	20(SDA),21(SCL)
3	LCD <i>Display</i>	20(SDA),21(SCL)
4	<i>Driver Relay</i> Indikator	D41,D43,D45
5	<i>Driver Relay</i> Motor untuk Kontrol Posisi	D33,D35,D37,D39
6	Sensor Fotodioda	A10,A11,A12
7	Sensor Potensiometer <i>Blower</i>	A8
8	Sensor Potensiometer <i>Valve</i>	A9
8	<i>Keypad</i> 3x4	D2,D3,D4,D5,D6,D7,D8
9	<i>Driver Relay Blower</i>	D51

3.1.2 Pengkabelan *Driver Relay* Motor DC

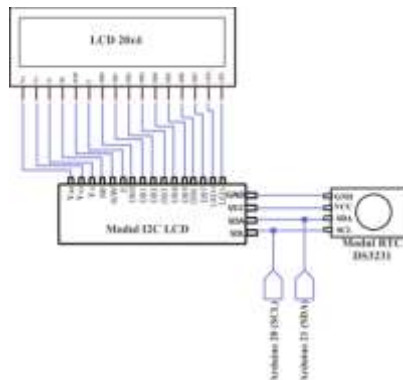
Driver relay 4 *channel* digunakan untuk menjalankan kedua motor DC agar dapat berjalan bolak balik dengan cara merubah polaritasnya dan mengatur agar *relay* tertentu yang aktif pada satu waktu. Pada kontak *relay* diberi tambahan dioda pengaman untuk mencegah masuknya arus balik dari eksitasi motor ke rangkaian kontrol utama.



Gambar 3.2 Pengkabelan Relay 4 Channel dengan Arduino

3.1.3 Koneksi I2C Bus RTC DS3231 dan LCD

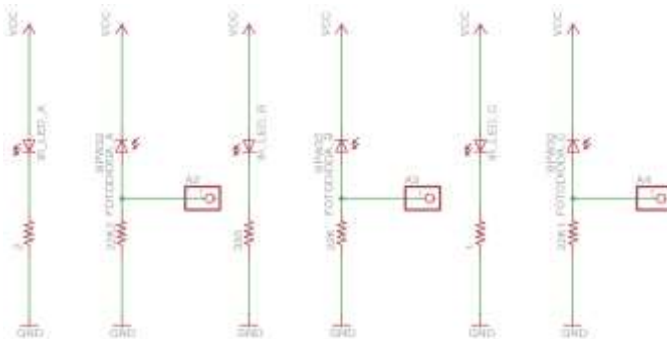
Modul Real Time Clock DS 3231 menggunakan koneksi *Inter Integrated Circuit* (I2C) untuk mengirim datanya ke mikrokontroler. Untuk LCD digunakan modul I2C sehingga dapat meringkas pin yang dipakai pada Arduino Mega. Digunakan metode I2C Bus untuk menyambung kedua modul I2C dalam satu sambungan data.



Gambar 3.3 Pengkabelan I2C Bus dari Modul RTC DS3231 dan Modul I2C LCD Ukuran 20x4

3.1.4 Perangkaian Sensor *Level* Fotodioda

Rangkaian fotodioda berfungsi untuk mendeteksi volume pakan yang tersisa didalam tangki. 3 buah fotodioda ditempatkan dalam ketinggian tertentu didalam tangki sehingga ketika fotodioda dan LED Inframerah terhalang pakan, arduino akan membaca nilai ADC yang berbeda dengan ketika kondisi fotodioda tidak terhalang. Kondisi logika di masing – masing fotodioda dimanfaatkan untuk mengindikasikan volume dengan indikator LED.

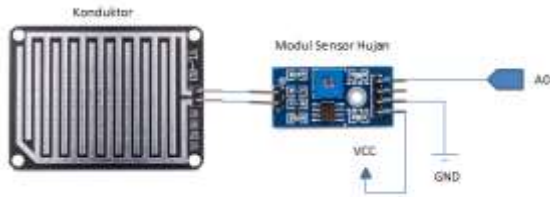


Gambar 3.4 Rangkaian Sensor Pendeteksi *Level* dengan Fotodioda ke Pin Analog Arduino Dan LED Inframerah

Penggunaan fotodioda yang dipasang secara berhadap-hadapan dengan LED inframerah berdasarkan penelitian Tugas Akhir Evi Siska Safarina dan Hendry Erwantono yaitu “*Rancang Bangun Lengan Robot 3 DOF Berbasis ATMEGA 328 untuk Memindahkan Biskuit*” memiliki pembacaan ADC serta tegangan sensor yang baik. Pada penelitian tersebut fotodioda digunakan untuk mendeteksi benda yang melintas di suatu konveyor berjalan.[3] Dengan prinsip kerja sensor yang hampir serupa yaitu mendeteksi benda, maka penggunaan sensor fotodioda ini sesuai dengan kebutuhan.

3.1.5 Rangkaian Sensor Hujan

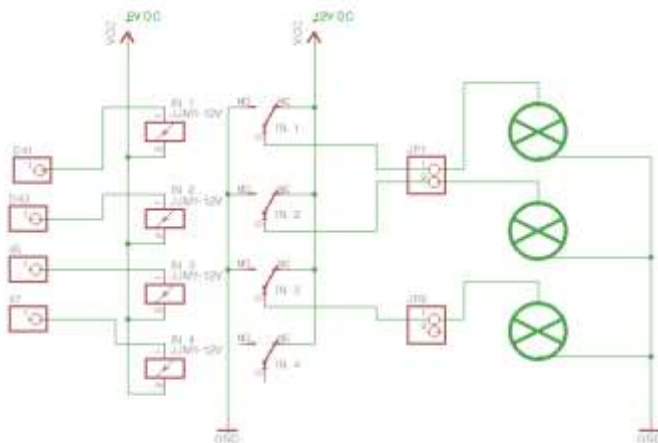
Sensor hujan memanfaatkan sebuah IC komparator untuk membandingkan tegangan yang melewati tembaga di *board* pcb dengan tegangan yang melewati potensiometer. Tegangan analog yang keluar dari pin AO modul sensor hujan kemudian masuk ke pin A0 arduino mega sebagai data ADC.



Gambar 3.5 Rangkaian Modul Sensor Hujan ke Arduino dan Penghantar

3.1.6 Perangkaian *Driver Relay* Lampu Indikator

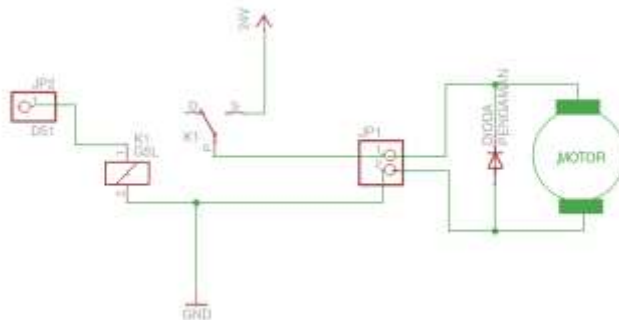
Lampu indikator yang ada di panel kontrol menggunakan sumber listrik DC sebesar 24 volt, oleh karena itu arduino tidak mampu menyalakan secara langsung tanpa bantuan aktuator berupa *driver relay*.



Gambar 3.6 Rangkaian *Driver Relay* untuk Lampu Indikator

3.1.7 Perangkaian *Driver Relay Blower*

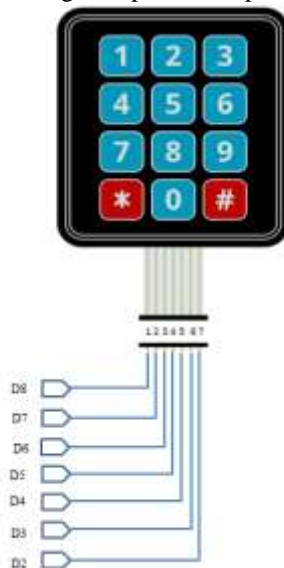
Blower merupakan komponen utama untuk melontarkan pakan yang jatuh dari valve menuju kearah kolam ikan. Untuk mengaktifkan *blower* sesuai keinginan, digunakan *driver relay 1 channel* tipe *active HIGH*.



Gambar 3.7 Perangkaian *Driver Relay* untuk *Blower Pakan*

3.1.8 Pengkabelan *Keypad 3x4*

Keypad 3x4 memiliki 7 pin yang terdiri dari 4 buah pin baris dan 4 3 buah pin kolom. Setiap pin yang ada pada *keypad* disambung pada pin *digital input* arduino. Penyambungan dapat dilihat pada gambar berikut :



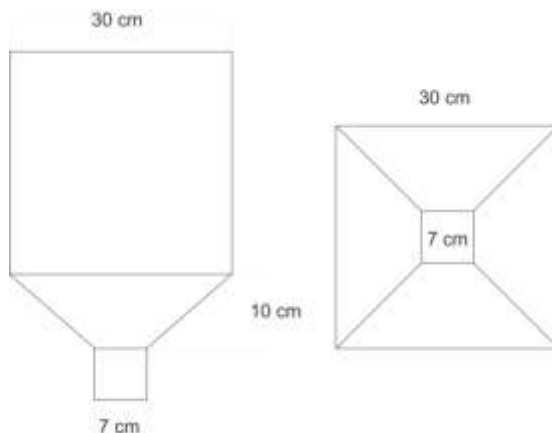
Gambar 3.8 Pengkabelan *Keypad 3x4* ke *Arduino Mega*

3.2 Pembuatan Perangkat Mekanik

Dalam pembuatan perangkat mekanik dari *feeder*, bagian – bagian alat dapat dikelompokkan menjadi 3, yaitu pembuatan tangki pakan, pembuatan mekanika *valve*, pembuatan mekanika *blower*, serta pembuatan rangka penahan *feeder*.

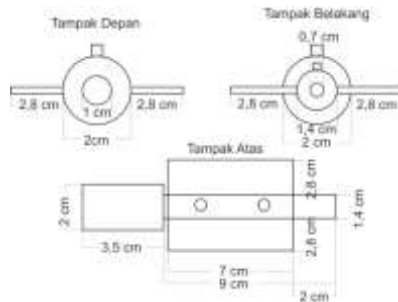
3.2.1 Pembuatan Tangki Pakan

Tangki untuk menampung pakan ikan terbuat dari besi galvanis. Lembaran plat besi dipotong, dibengkokkan, serta dilas untuk mendapatkan bentuk yang sesuai dengan desain. Pada bagian bawah tangki, bentuknya dibuat mengerucut untuk memusatkan jatuhnya pakan pada satu titik. Lubang bagian bawah diberi perpanjangan untuk memberi tempat bagi pemasangan *valve outlet* pakan.



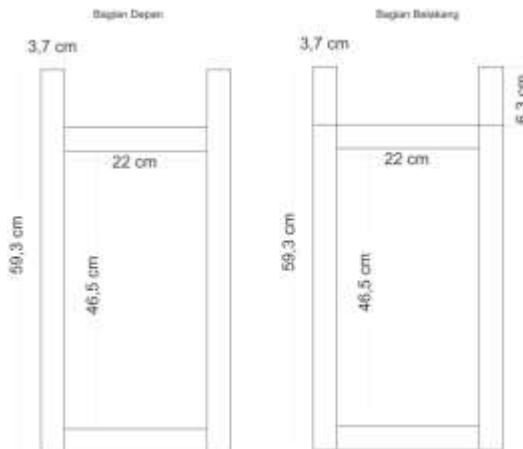
Gambar 3.9 Desain Ukuran Tangki

Gambar 3.10 berikut adalah rancangan *valve* buka-tutup yang dipasang di bagian bawah corong tangki sehingga dapat mengatur keluar tidaknya pakan ikan dari tangki. Bagian kiri *valve* (tampak atas) merupakan tempat untuk mengkoppel poros motor DC *gearbox*, sedangkan bagian ujung kanan digunakan untuk mengkoppel sensor posisi potensiometer sehingga kedua poros akan berputar bersamaan. Bagian tengahnya diberi sebuah plat yang disisipkan diantara as dan dikunci dengan baut, fungsinya untuk menahan pakan dari tangki sesuai dengan ukuran ujung bawah tangki.



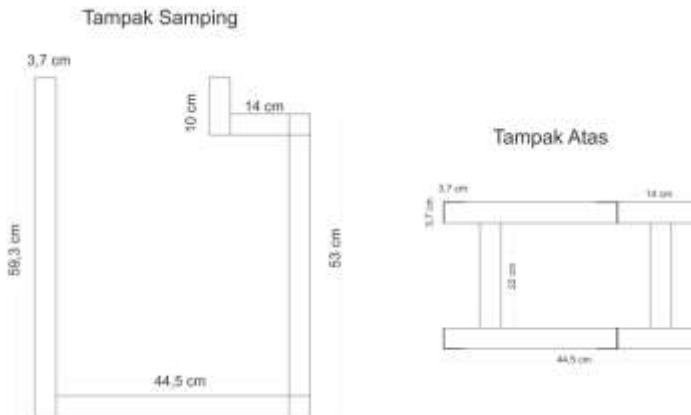
Gambar 3.10 Desain *Valve Output* Pakan Ikan

Tangki pakan memiliki mekanisme untuk membuka dan menutup *valve* sehingga pakan ikan bisa keluar secara otomatis. Untuk memenuhi spesifikasi ini, digunakan sebuah motor DC untuk memutar *valve output* sampai posisi tertentu. Motor DC dilengkapi dengan *gearbox* konfigurasi *worm gear* untuk menahan beban mekanis walaupun motor sedang tidak ter-eksitasi.



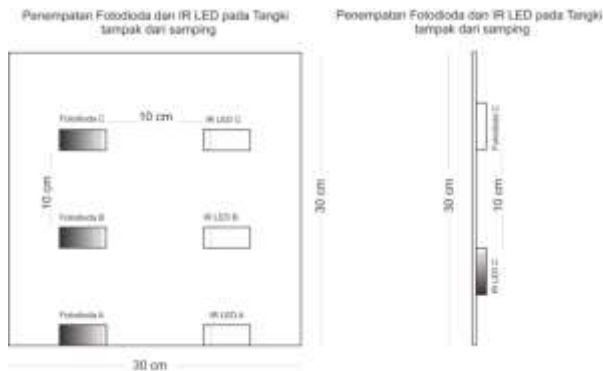
Gambar 3.11 Bagian Depan dan Belakang Rangka Penahan Tangki

Gambar 3.11 dan Gambar 3.12 adalah desain rangka penahan tangki agar dapat berdiri kokoh diatas *blower* sehingga nantinya pakan dapat terjatuh dari dalam tangki menuju mekanisme penyebaran oleh *blower*. Rangka terbuat dari plat besi siku dan lurus yang dilas dan disambung sedemikian rupa sehingga dapat menahan keseluruhan komponen mekanis dari alat.



Gambar 3.12 Bagian Samping dan Atas Rangka Penahan Tangki

Gambar 3.13 berikut merupakan desain penempatan sensor fotodiode dan LED inframerah pada dinding tangki. Masing – masing jumlahnya 3 buah dan ditempatkan secara berhadap – hadapan. Tujuannya adalah untuk memaksimalkan sensitivitas pembacaan sinar inframerah oleh fotodiode. Jarak tiap fotodiode dengan IR LED adalah 10 cm sedangkan jarak antar komponen yang sama adalah 10 cm.



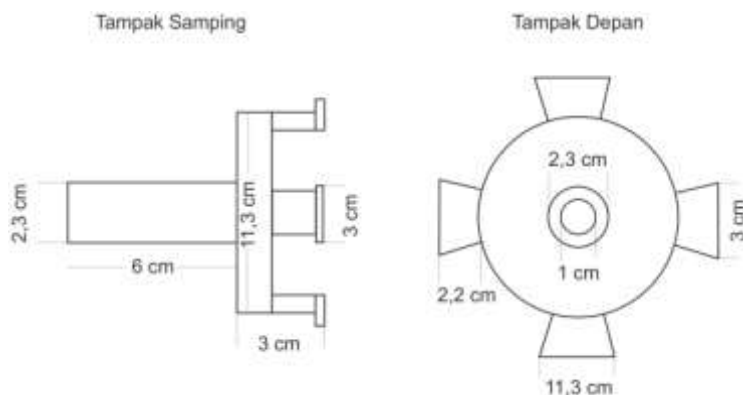
Gambar 3.13 Penempatan Sensor Fotodiode Dan LED Inframerah Pada Dinding Tangki Pakan

3.2.2 Pembuatan Mekanisme Blower

Blower merupakan mesin penggerak utama bagi alat untuk menyebar pakan sehingga dapat menjangkau jarak yang lebih jauh. Tentunya sekedar meniup pakan saja tidak cukup untuk memenuhi spesifikasi yang kita harapkan. Diperlukan pengaturan sudut tiup *blower* sehingga dengan arah sebar yang berubah ubah, persebaran pakan dapat merata. Ini dikarenakan ikan gurami memiliki sifat yang tidak terlalu agresif ketika makan, bila pakan tidak merata dikhawatirkan pertumbuhan ikan tidak merata juga.

Motor DC digunakan untuk menggerakkan *chassis blower* sehingga arah angin yang dihasilkan juga berubah. Motor DC yang digunakan memiliki *gearbox* tambahan untuk memberi keuntungan mekanis bagi sistem. Dengan mekanisme *worm gear*, motor tidak akan terputar porosnya ketika ditekan oleh beban.

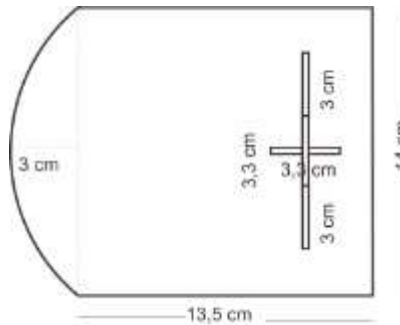
Karena motor DC tidak dapat diatur secara langsung posisinya seperti pada motor *servo* dan motor *stepper*, maka kita memerlukan mekanisme tambahan untuk mengetahui posisi sudut poros motor. Potensiometer dikopel dengan poros motor sehingga pergerakan poros akan menghasilkan perubahan resistansi yang dapat dideteksi oleh sistem. Gambar 3.14 berikut merupakan rumah poros penahan *blower* dimana *blower* akan dipasang dan dikopel dengan poros motor DC *gearbox* sehingga posisinya akan mengikuti pergerakan poros motor.



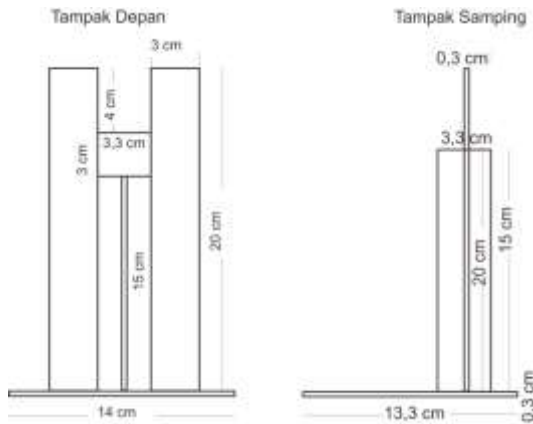
Gambar 3.14 Bagian Poros Penahan *Blower*

Gambar 3.14 dan Gambar 3.15 berikut merupakan rangka penahan motor DC *gearbox* dan poros *blower*. Rangka ini juga berfungsi sebagai penyambung mekanisme *blower* pada rangka tangki pakan. Agar *blower*

dan motor dapat terpasang dengan kuat, sekrup dan baut tambahan dipasang diantara rumah poros dengan *blower* dan motor.



Gambar 3.15 Bagian Penahan Motor DC Tampak Dari Atas



Gambar 3.16 Bagian Penahan Motor DC dari Depan dan Samping

Setelah mekanisme *blower* jadi, kemudian keseluruhan bagian digabung dengan rangka penahan tangki dan dibaut agar tidak mudah goyah akibat getaran *blower* dan motor DC penahannya.

3.3 Pembuatan Perangkat Lunak

Dalam pembuatan perangkat lunak terdapat beberapa program yang harus disusun agar mekanisme kerja *feeder* dapat berjalan dengan baik. Tahapan yang dilalui diantaranya adalah pembuatan *flowchart* program, pembuatan program penggerak mekanisme *blower*, pembuatan program

penggerak mekanisme *valve* pakan, pembuatan program indikator pakan, serta pembuatan program antarmuka bagi pengguna.

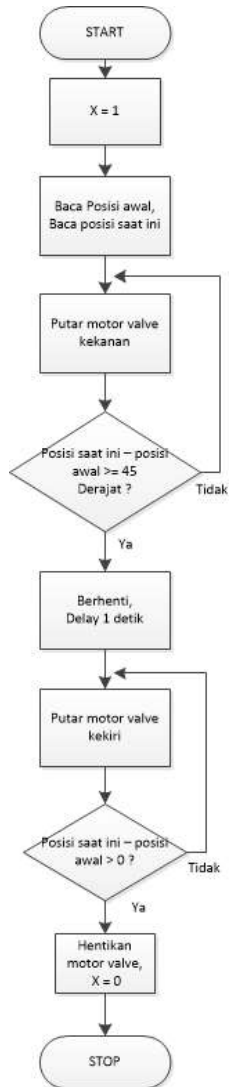
3.3.1 Pembuatan *Flowchart* Program

Flowchart adalah bagan yang memperlihatkan urutan dan hubungan antar proses beserta instruksinya. Bagan ini dinyatakan dengan simbol. Dengan demikian setiap simbol menggambarkan proses tertentu. Sedangkan hubungan antar proses digambarkan dengan garis penghubung. *Flowchart* ini merupakan langkah awal pembuatan program. Dengan adanya *flowchart* urutan poses kegiatan menjadi lebih jelas.

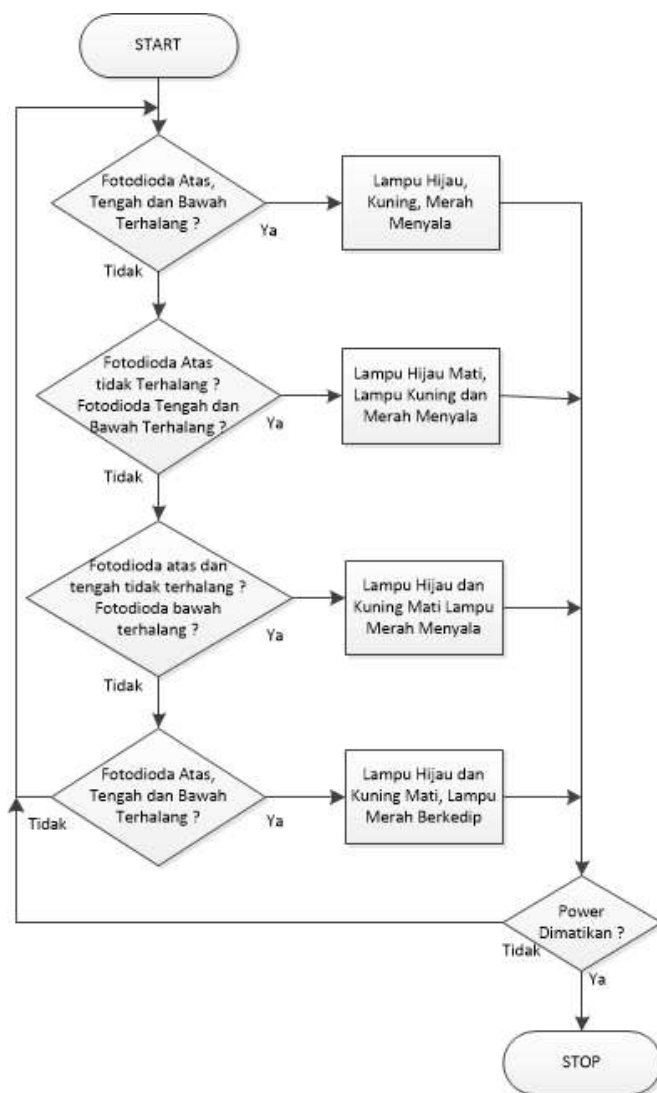
Untuk pengolahan data dengan komputer, dapat dirangkum urutan dasar untuk pemecahan suatu masalah, yaitu;

- *START*: berisi instruksi untuk persiapan peralatan yang diperlukan sebelum menangani pemecahan masalah.
- *READ*: berisi instruksi untuk membaca data dari suatu peralatan.
- *PROCESS*: berisi kegiatan yang berkaitan dengan pemecahan persoalan sesuai dengan data yang dibaca.
- *WRITE*: berisi instruksi untuk merekam hasil kegiatan ke peralatan *output*.
- *END*: mengakhiri kegiatan pengolahan.[3]

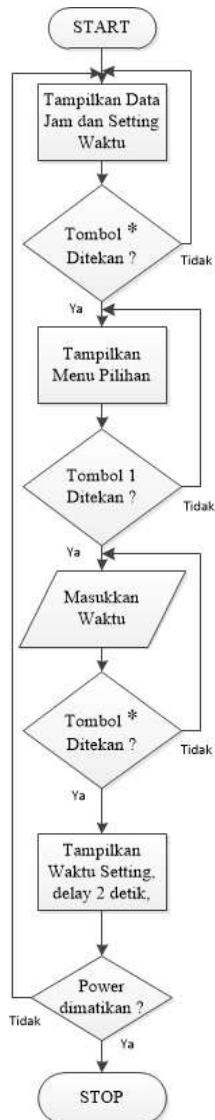
Flowchart program dari Tugas Akhir ini terbagi menjadi beberapa bagian sesuai dengan alur program yang sedang berjalan.



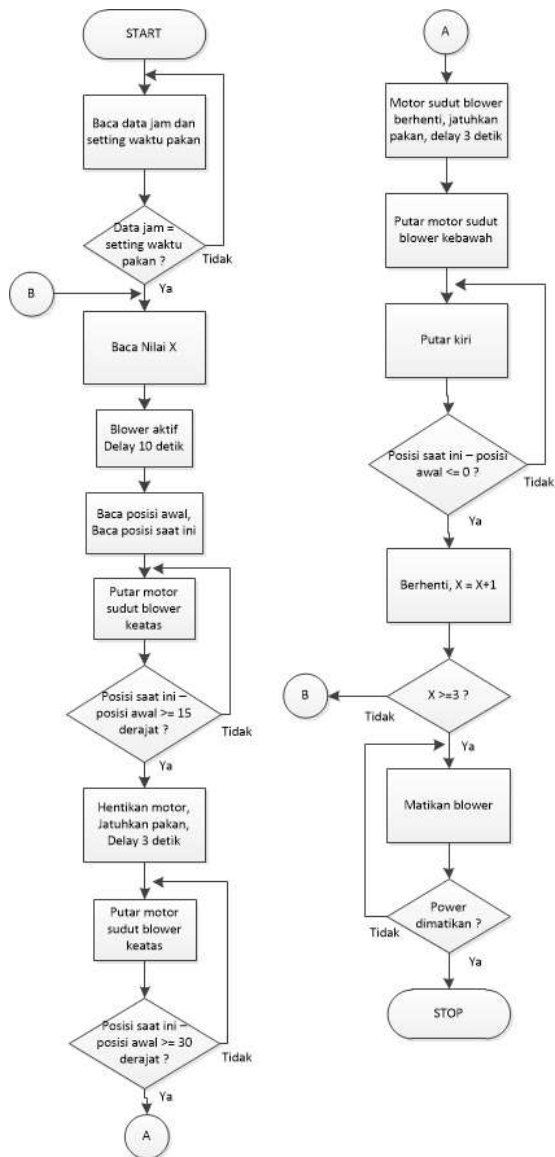
Gambar 3.17 Flowchart Program untuk Menjatuhkan Pakan Melalui Valve Tangki



Gambar 3.18 Flowchart Porgram Indikator Tangki



Gambar 3.19 Flowchart Program untuk Mengatur Waktu Pakan Melalui Antarmuka LCD dan Keypad



Gambar 3.20 Flowchart Program Pemberian Pakan

3.3.2 Pendefinisian Pin dan Deklarasi Variabel

```
#include <DS1307.h> //Memanggil RTC library
#include <Keypad.h> //library Keypad
#include <LiquidCrystal_I2C.h> //library LCD I2C
]

int posisiValve;
int valveDulu;
int posisi;
int posisiSebelum;
const int K1merPin = 31;
const int K1L = 32;
const int K2L = 33;
const int K3L = 37;
const int K4L = 39;
const int potensiPin = A5;
const int valvePin = A3;
const int indikator_A = 45;
const int indikator_B = 43;
const int indikator_C = 41;
const int sensor_A = A12;
const int sensor_B = A11;
const int sensor_C = A10;
int ADC_A;
int ADC_B;
int ADC_C;
unsigned long Dulu = 0;
const long relais = 250;
int switching = LOW;
```

Gambar 3.21 Bagian Pendefinisian Pin dan Deklarasi Variabel

Agar program dapat berjalan dengan baik dan lebih mudah dikoreksi, pada bagian awal program terlebih dahulu didefinisikan pin-pin yang digunakan dan dideklarasikan variabel maupun konstanta yang nantinya akan digunakan oleh program.

Variabel dengan tipe data *integer* dapat menyimpan nilai sebesar 16 Bit. *Const* merupakan deklarasi untuk konstanta dan digunakan untuk mendefinisikan pin pada arduino. Tipe data *long* seperti pada variabel dulu digunakan untuk menyimpan nilai waktu di segmen millis() karena tipe *integer* tidak cukup untuk menampungnya.

Variable posisi bertipe *integer* seperti posisiValve, valveDulu, posisi, dan posisiSebelum digunakan untuk menyimpan data ADC dari potensiometer. IN1 hingga IN4 merupakan konstanta untuk pin di arduino mega yang digunakan untuk mengontrol motor.

```
char keypad[];
const byte ROWS = 4; //jumlah baris keypad
const byte COLS = 3; //jumlah kolom keypad
char keys[ROWS][COLS] = {
  {'1','2','3'},
  {'4','5','6'},
  {'7','8','9'},
  {'*','0','#'}};
byte rowPins[ROWS] = {0,7,6,5}; //pin baris untuk keypad
byte colPins[COLS] = {4,3,2}; //pin kolom untuk keypad
Keypad myKeypad( matrixKeymap(keys, rowPins, colPins, ROWS, COLS));

LiquidCrystal_I2C lcd(0x3F,2,1,0,4,5,6,7,3, POSITIVE); //0x3F = alamat I2C ICD
```

Gambar 3.22 Pendefinisian Keypad dan LCD Display

Pada gambar 3.22 dapat kita amati pen-settingan awal untuk *keypad* 3x4 dan LCD *display*. Tombol yang ada pada *keypad* didefinisikan pada sebuah matriks baris dan kolom dengan tipe data *char*. Selanjutnya pin arduino yang digunakan *keypad* didefinisikan. Untuk mempermudah penggunaannya *library* keypad.h sudah menyediakan perintah perintah yang membantu mempermudah pen-settingan *keypad*, seperti pada proses *mapping*.

Kemudian terdapat perintah pendefinisian LCD *display* yang menggunakan koneksi I2C. Alamat LCD yaitu 0x3F didefinisikan agar tidak rancu dengan RTC yang juga menggunakan koneksi I2C pada arduino.

3.3.3 Bagian Program untuk Pengaturan Awal

```
void setup() {
  rtc.begin();           //memulai koneksi i2c dengan RTC
  Serial.begin(9600);    //memulai koneksi serial
  lcd.begin(20,4);       //memulai koneksi LCD
  sensor.begin();        //memulai koneksi sensor
  pinMode(pintRajam, INPUT); //Inialisasi input pin Rajam
  pinMode(pompa, OUTPUT); //Inialisasi output pompa
  pinMode(blower, OUTPUT);
  pinMode(indicator_A, OUTPUT);
  pinMode(indicator_B, OUTPUT);
  pinMode(indicator_C, OUTPUT);
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);
  pinMode(blowerPin, OUTPUT);
  Serial.begin(9600);
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, HIGH);
  digitalWrite(IN3, HIGH);
  digitalWrite(IN4, HIGH);
}
```

Gambar 3.23 Bagian Pengaturan Awal Program

Void pada Arduino digunakan untuk membuat fungsi atau sub program yang dapat dipanggil. *Void setup* pada IDE Arduino dieksekusi diawal dengan perulangan satu kali saja. Perintah *rtc.begin* digunakan untuk mengaktifkan fungsi rtc. Sedangkan perintah *Serial.begin* digunakan untuk mengaktifkan komunikasi serial dengan baud rate 9600.

Pada Gambar 3.23 juga terdapat perintah *pinMode*, perintah ini digunakan untuk mendeklarasikan pin pada arduino sebagai *input* atau *output* seperti pada gambar. Perintah *digitalWrite* digunakan untuk memberi *output* pada pin *digital* baik sinyal *HIGH* (5 Volt) maupun sinyal *LOW* (0 Volt).

Dalam bagian program diatas pin IN1 hingga IN4 didefinisikan sebagai output dan IN1 sampai IN3 diberi sinyal *HIGH*. Sedangkan IN4

diberi sinyal *LOW*. Keempat pin ini akan men-drive *driver relay* untuk menggerakkan motor. Didalam segmen diatas terdapat tambahan variabel bertipe *integer* untuk memberi kondisi pada jalannya program.

3.3.4 Bagian Program untuk Membaca Posisi dan Mengatur Motor

```

int x = 20;
digitalWrite(blowerPin, HIGH);
e=millis();
for(;;){
  while((f-e) <= 8000){
    f=millis();
  }
  pakan();
  //START MOTOR
  posisiSebelum = analogRead(potensiPin);
  posisi = analogRead(potensiPin);
  while((posisi - posisiSebelum) <= 80){
    putarKanan();
    posisi = analogRead(potensiPin);
  }
}

```

Gambar 3.24 Bagian Program Pembacaan Posisi dan *Starting* Motor

Pada gambar 3.24, program masuk didalam *void* tersendiri, artinya susunan program dapat dipanggil untuk berjalan secara instan. Pada bagian awal terdapat pemberian nilai pada variabel *x*, dimana hal ini digunakan untuk memberikan kondisi pada program dan mengendalikan perulangan program.. Variabel ini akan dibahas kembali pada bagian akhir sub program pakan. Kemudian pin *relay* yang men-drive *blower* diaktifkan. Maksud dari perintah *analogRead* adalah untuk membaca data analog yang ada pada pin yang dideklarasikan sebagai *potensiPin*. Kemudian nilai yang didapat dimasukkan dalam variabel *posisi*.

Selanjutnya fungsi *while* digunakan untuk menjalankan suatu perintah selama kondisi yang disyaratkan terpenuhi. Disini jika nilai variabel *posisi* dikurangi dengan variabel *posisi* sebelum kurang dari nilai yang ditentukan maka perintah dalam kurung kurawal akan dijalankan yaitu *putarKanan* dan pembacaan *posisi*.

3.3.5 Bagian Program Untuk Memberi Jeda Berhenti

```

berhenti();
pakan();
//Delay 2 detik
e=millis();
f=millis();
while((f-e) <= 2000){
  f=millis();
}

```

Gambar 3.25 Bagian Program untuk Memberi Jeda dengan *Millis()*

Dalam segmen ini pergerakan sudut dihentikan dan mekanisme untuk menjatuhkan pakan dari tangki dijalankan. Disini variabel a dan b diisi dengan nilai perintah `millis()`. `millis()` adalah *timer* internal arduino yang digunakan untuk menghitung waktu sejak arduino di-*reset* dalam satuan milidetik. Saat nilai b dikurangi nilai a kurang dari sama dengan nilai yang ditentukan maka b tetap berisi nilai `millis()`.

3.3.6 Bagian Program Untuk Mengembalikan Posisi Sudut pada Titik Awal dan Berhenti

```
while((posisi-posisiSebelum) >= 0){
    putarKiri();
    posisi = analogRead(potensioPin);
}
//stage 3
berhenti();
x=0;
digitalWrite(blowerPin, LOW);
}
}
```

Gambar 3.26 Bagian Program untuk Mengembalikan Posisi Motor dan Berhenti

Pada segmen program ini apabila posisi dikurangi dengan `posisiSebelum` lebih dari nol maka motor pengatur sudut *blower* akan berputar kekiri atau berlawanan dengan arah putaran sebelumnya. Nilai posisi masih dipantau seperti sebelumnya menggunakan perintah `analogRead` pada `potensioPin`.

Apabila kondisi posisi motor sudah kurang dari nol atau tidak memenuhi persyaratan dari *while* maka motor berhenti dan kondisi x menjadi bernilai nol. Hal ini untuk menghentikan perulangan program pemberian pakan. Selanjutnya motor *blower* untuk meniup pakan dimatikan dengan perintah `digitalWrite` pada pin pengatur *relay blower*.

Bagian program ini dapat dimodifikasi untuk mengatur perulangan proses peniupan pakan. Dikarenakan variabel x menjadi kunci untuk menghentikan proses pemberian pakan.

3.3.7 Bagian Program untuk Membuat Fungsi

```
void berhenti() {  
    digitalWrite(IN3, HIGH);  
    digitalWrite(IN4, HIGH);  
}  
  
void putarKanan() {  
    digitalWrite(IN3, HIGH);  
    digitalWrite(IN4, LOW);  
}  
  
void putarKiri() {  
    digitalWrite(IN3, LOW);  
    digitalWrite(IN4, HIGH);  
}
```

Gambar 3.27 Bagian Program untuk Membuat Fungsi

Fungsi digunakan untuk memanggil keseluruhan perintah yang terkandung didalamnya ketika diaktifkan. Dengan fungsi penulisan program akan menjadi lebih ringkas karena tidak perlu lagi menulis perintah yang sama berulang-ulang. Disini perintah *void* digunakan untuk mendeklarasikan sebuah fungsi. Fungsi berhenti, putarKanan, dan putarKiri digunakan untuk men-*drive relay* pengatur motor agar dapat berjalan sesuai keinginan dengan menggunakan logika *HIGH* dan *LOW* berdasarkan sambungan pengkabelan pada papan rangkaian.

3.3.8 Bagian Program Logika Indikator

```
if(ADC_A > 200 && ADC_B > 200 && ADC_C > 200) {  
    digitalWrite(indicator_A, LOW);  
    digitalWrite(indicator_B, LOW);  
    digitalWrite(indicator_C, LOW);}   
if(ADC_A < 200 && ADC_B > 200 && ADC_C > 200) {  
    digitalWrite(indicator_A, HIGH);  
    digitalWrite(indicator_B, LOW);  
    digitalWrite(indicator_C, LOW);}   
if(ADC_A < 200 && ADC_B < 200 && ADC_C > 200) {  
    digitalWrite(indicator_A, HIGH);  
    digitalWrite(indicator_B, HIGH);  
    digitalWrite(indicator_C, LOW);}   
if(ADC_A < 200 && ADC_B < 200 && ADC_C < 200) {  
    unsigned long sekarang = millis();  
    digitalWrite(indicator_A, HIGH);  
    digitalWrite(indicator_B, HIGH);
```

Gambar 3.28 Bagian Program Logika untuk Indikator

Indikator tangki berupa lampu 3 warna aktif berdasarkan pembacaan sensor fotodioda yang terpasang pada tangki. Data yang didapat dari

fotodioda diolah oleh arduino seperti pada program diatas. Jika seluruh fotodioda terhalang, mengindikasikan bahwa stok pakan didalam tangki masih banyak. Oleh karena itu lampu merah kuning dan hijau menyala. Kemudian bila fotodioda bagian atas tidak terhalang maka pakan telah berkurang hingga kurang dari *level* tersebut. Oleh karena itu lampu hijau mati dan lampu hijau serta merah menyala. Demikian seterusnya hingga *level* pakan kurang dari fotodioda paling bawah, artinya stok pakan sudah sangat tipis. Karenanya lampu indikator merah menyala berkedip. Dalam bagian program ini digunakan ADC dari fotodioda untuk mendeteksi terhalang tidaknya fotodioda oleh pakan.

BAB IV

PENGUJIAN DAN PENGAMBILAN DATA

Untuk mengetahui bahwa alat telah bekerja dengan benar maka perlu dilakukan pengujian alat yang meliputi pengujian perangkat keras dan pengujian perangkat lunak. Pengujian yang dilakukan pada peralatan untuk mengetahui kesesuaian antara teori dengan hasil perancangan, yaitu dengan mengetahui hasil pengukuran pada setiap perangkat yang telah dibuat. Alat yang telah selesai dapat dilihat pada Gambar 4.1



Gambar 4.1 Gambar Keseluruhan Alat Disertai Kolam dan *Box* Kontroler

4.1 Data Pengujian Peralatan

Untuk memastikan bahwa tiap bagian peralatan dapat bekerja sebagaimana fungsinya maka dilakukan pengujian peralatan.

4.1.1 Pengujian LCD 20x4

Untuk menguji LCD 20x4 ini digunakan program sederhana untuk menampilkan karakter pada tiap baris LCD. Mula mula diuji apakah LCD dapat menampilkan karakter sesuai dengan lokasi yang dimaksud perintah *lcd.setCursor*. Program yang digunakan adalah sebagai berikut :

```

#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x3F, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);
void setup() {
    lcd.begin(20, 4);
}

void loop() {
    lcd.setCursor(0, 0); //baris pertama
    lcd.print("Baris 1");
    lcd.setCursor(0, 1); //baris kedua
    lcd.print("Baris 2");
    lcd.setCursor(0, 2); //baris ketiga
    lcd.print("Baris 3");
    lcd.setCursor(0, 3); //baris keempat
    lcd.print("Baris 4");
}

```

Gambar 4.2 Program pengujian fungsi LCD

Hasilnya, LCD dapat menampilkan karakter pada tempat yang diinginkan, seperti pada gambar berikut :



Gambar 4.3 Tampilan Layar LCD dari Pengujian Program

Selanjutnya diuji apakah seluruh bagian LCD dapat menampilkan karakter sebanyak ukurannya yaitu 20 karakter tiap baris. Dikhawatirkan terjadi kerusakan pada salah satu karakter LCD oleh karena itu pengujian dilakukan dengan menampilkan karakter ke setiap baris dan kolom LCD.

```

#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x3F, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);
void setup() {
    lcd.begin(20, 4);
}

void loop() {
    lcd.setCursor(0, 0); //baris pertama
    lcd.print("ABCDEFGHIJKLMNPQRST");
    lcd.setCursor(0, 1); //baris kedua
    lcd.print("ABCDEFGHIJKLMNPQRST");
    lcd.setCursor(0, 2); //baris ketiga
    lcd.print("ABCDEFGHIJKLMNPQRST");
    lcd.setCursor(0, 3); //baris keempat
    lcd.print("ABCDEFGHIJKLMNPQRST");
}

```

Gambar 4.4 Program untuk Menguji Karakter LCD

Hasilnya LCD dapat menampilkan keseluruhan karakter dengan jumlah maksimalnya yaitu 20 karakter tiap baris seperti pada gambar. Artinya LCD dalam keadaan baik dan siap digunakan.



Gambar 4.5 Tampilan Layar LCD Hasil Pengujian Karakter

4.1.2 Pengujian Keypad

Untuk menguji *keypad* ini digunakan program sederhana untuk membaca karakter dari tiap tombol yang ada pada *keypad* dan menampilkannya pada *serial monitor* arduino.

```
#include <Keypad.h>
const byte ROWS = 4;
const byte COLS = 3;
char keys[ROWS][COLS] = {
  {'1','2','3'},
  {'4','5','6'},
  {'7','8','9'},
  {'*','0','#'}
};

byte rowPins[ROWS] = {2, 7, 6, 5};
byte colPins[COLS] = {4, 3, 2};
Keypad customKeypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS);

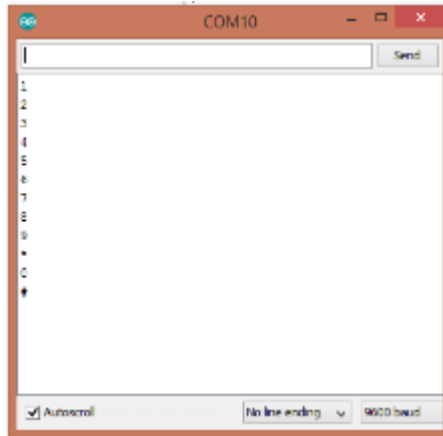
void setup() {
  Serial.begin(9600);
}

void loop() {
  char customKey = customKeypad.getKey();

  if (customKey) {
    Serial.println(customKey);
  }
}
```

Gambar 4.6 Program Pengujian Keypad

Dari hasil pengujian, seluruh tombol pada *keypad* berfungsi dengan baik karena karakter yang dimaksud muncul pada *serial monitor* ketika tombol *keypad* ditekan.



Gambar 4.7 Tampilan pada *Serial Monitor* Saat Masing-Masing Tombol Ditekan

4.1.3 Pengujian Koneksi Sensor

Untuk menguji apakah masing masing sensor telah terhubung dan dibaca kondisinya melalui Arduino maka digunakan pengujian dengan program sederhana.

```

Serial.begin(9600);

// the 'loop' routine runs over and over again forever
void loop() {
  // read the input on analog pin 0:
  int fdiadaC = analogRead(A10);
  int fdiadaB = analogRead(A11);
  int fdiadaA = analogRead(A12);
  int potensioC = analogRead(A9);
  int potensioB = analogRead(A8);
  int hujan = analogRead(A0);
  // print out the value you read:
  Serial.print(fdiadaA);
  Serial.print("\n-----");
  Serial.print(fdiadaB);
  Serial.print("\n-----");
  Serial.print(fdiadaC);
  Serial.print("\n-----");
  Serial.print(potensioA);
  Serial.print("\n-----");
  Serial.print(potensioB);
  Serial.print("\n-----");
  Serial.print(hujan);
  Serial.println();
  delay(100); // delay in between reads for stability
}

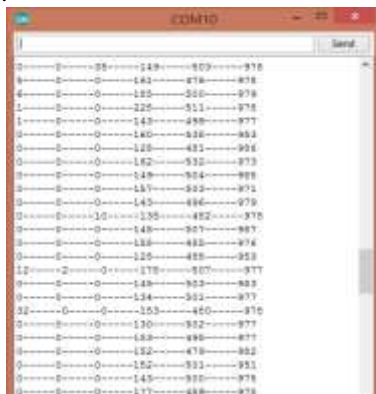
```

Tabel 4.1 Hasil Pengujian Sensor – Sensor yang Ada

No	Nama Sensor	Kondisi
1	Fotodiode A	Data Terbaca
2	Fotodiode B	Data Terbaca

3	Fotodioda C	Data Terbaca
4	Potensiometer Atas	Data Terbaca
5	Potensiometer Bawah	Data Terbaca
6	Sensor Hujan	Data Terbaca

Dari hasil pengujian arduino telah dapat membaca nilai data sensor melalui ADC (*Analog to Digital Converter*). Hasil pembacaan sensor adalah sebagai berikut :



Gambar 4.8 Pembacaan Nilai Sensor pada Serial Monitor

4.1.4 Pengujian Motor DC

Untuk menguji apakah motor yang terdapat pada alat dapat bekerja dengan baik maka dilakukan pengujian dengan mencatu motor baik *forward* maupun *reverse* pada sumber tegangan DC sehingga dapat diketahui apakah motor mengalami kerusakan.

Tabel 4.2 Tabel Pengujian Motor DC

No	Motor yang Diuji	Kondisi
1	Motor Sudut Pakan	Baik
2	Motor Sudut <i>Valve</i>	Baik
3	Motor <i>Blower</i>	Baik

4.1.5 Pengujian *Real Time Clock*

Untuk menguji apakah RTC dapat memberikan data waktu yang tepat bagi mikrokontroler dilakukan pengujian dengan menampilkan pembacaan waktu pada *LCD Display* dan dibandingkan dengan waktu pada komputer.

```

#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <DS1307.h>
#include <RTClib.h>
LiquidCrystal_I2C lcd(SS, RW, RS);
void setup() {
  lcd.begin(16,4);
  lcd.begin(0,0);
  //rtc.setTime(10, 36, 11); // Set the time to 10:36:11 (4th March)
  //rtc.setDate(7, 2017); // Set the date as January 1st, 2017
}

void loop() {
  lcd.setCursor(0, 0);
  lcd.print(rtc.getTime());
  lcd.print(" ");
  lcd.print(" ");
  lcd.print(rtc.getDate());
  lcd.setCursor(0, 1);
}

```

Gambar 4.9 Program yang Digunakan untuk Pengujian RTC



Gambar 4.10 Tampilan Waktu Hasil Pengujian RTC pada LCD Display



Gambar 4.11 Waktu pada Jendela *Date and Time Windows*

Dari hasil pengujian, RTC dapat menampilkan waktu sesuai setting waktu yang dilakukan sebelumnya. Akan tetapi terdapat selisih waktu keterlambatan selama 30 detik dari waktu pada jam komputer yang kami anggap tepat. Hal ini disebabkan adanya selisih ketika program sedang di-*upload* dan ketepatan saat meng-*upload* program juga menjadi salah satu faktor kepresisian waktu RTC.

4.1.6 Pengujian *Relay*

Untuk menguji apakah *relay* dapat mengontak atau tidak ketika diberi sinyal *HIGH* dan *LOW* dilakukan pengujian dengan program sederhana

pada masing masing *relay*. Hasil pengujian dapat dilihat pada Tabel 4.3 berikut :

Tabel 4.3 Tabel Kondisi *Relay* Saat Diuji

No	<i>Relay</i> yang Diuji	Kondisi
1	<i>Relay</i> A1	Baik
2	<i>Relay</i> A2	Baik
3	<i>Relay</i> A3	Baik
4	<i>Relay</i> A4	Baik
5	<i>Relay</i> B1	Baik
6	<i>Relay</i> B2	Baik
7	<i>Relay</i> B3	Baik
8	<i>Relay</i> B4	Baik
9	<i>Relay</i> C (<i>Blower</i>)	Baik
10	<i>Relay</i> D (<i>Pompa</i>)	Baik

Relay dianggap bekerja dengan baik apabila kontaknya aktif sesuai dengan program pengujian. Hasilnya masing masing kontak *relay* dapat berfungsi dengan baik dengan menyala dan mati selama 1 detik secara bergantian.

4.1.7 Pengujian Indikator Tangki

Untuk menguji apakah lampu indikator pada *control box* dapat berfungsi dengan baik, digunakan program sederhana untuk menyalakan lampu dengan kontak *relay*. Hasil pengujian dapat dilihat pada Tabel 4.4 berikut :

Tabel 4.4 Tabel Kondisi Tiap Lampu Saat Diuji

No	Warna Lampu	Kondisi
1	Lampu Hijau	Baik
2	Lampu Kuning	Baik
3	Lampu Merah	Baik

Selanjutnya lampu diprogram untuk menyala dan mati sesuai dengan kondisi isi tangki, sehingga nyala lampu menjadi indikator tingkat isi pakan ikan yang ada didalam tangki. Hasil pengujian dapat dilihat pada Tabel 4.5 berikut :

Tabel 4.5 Hasil Pengujian Program Indikator Tangki

No	Fotododa Atas	Fotodioda Tengah	Fotodioda Bawah	Lampu Hijau	Lampu Kuning	Lampu Merah
1	Terhalang	Terhalang	Terhalang	Nyala	Nyala	Nyala
2	Tidak Terhalang	Terhalang	Terhalang	Mati	Nyala	Nyala
3	Tidak Terhalang	Tidak Terhalang	Terhalang	Mati	Mati	Nyala
4	Tidak Terhalang	Tidak Terhalang	Tidak Terhalang	Mati	Mati	Berkedip



Gambar 4.12 Kondisi Ketiga Lampu Menyala



Gambar 4.13 Kondisi Satu Lampu Mati



Gambar 4.14 Kondisi Dua Lampu Mati



Gambar 4.15 Kondisi Dua Lampu Mati dan Satu Lampu Berkedip

4.2 Pengukuran dan Pengujian Sensor *Level* Tangki

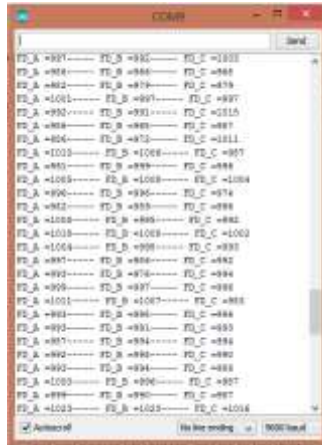
Tujuan dari pengujian sensor *level* adalah untuk mengetahui tingkat perubahan tegangan dan ADC ketika fotodioda dan LED inframerah terhalang pakan dan tidak terhalang.

Cara pengambilan data yang digunakan adalah memprogram Arduino dan membaca nilai ADC melalui komunikasi *serial*. Dalam program, apabila sensor terhalang oleh pakan maka lampu indikator aktif, sedangkan jika tidak ada pakan yang terdeteksi maka lampu indikator akan non-aktif. Pengujian dilakukan dengan mengukur tegangan pada pin data dan pin *ground* rangkaian fotodioda dan dihadapkan ke LED inframerah yang dinyalakan pada jarak 10 cm. Untuk menguji saat terhalang, digunakan pakan ikan pelet jenis 781-3 produksi PT Charoen Pokphand Prima.

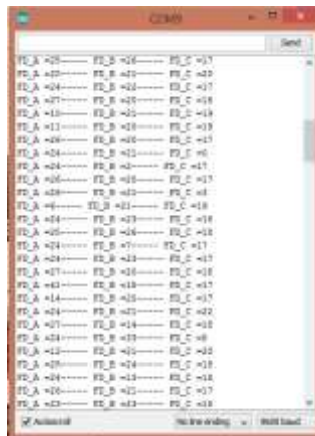
Hasil pengujian dapat diamati pada Tabel 4.6 berikut :

Tabel 4.6 Pengukuran Sensor Fotodioda

Sensor	Tegangan Saat Tidak Ada Pakan	Tegangan Saat Ada Pakan
A	0,132 V	4,86 V
B	0,125 V	4,8 V
C	0,109 V	4,89 V
Rata-Rata	0,122 V	4,82 V



Gambar 4.16 Pembacaan ADC Fotodiode Pada Saat Terdapat Pakan



Gambar 4.17 Pembacaan ADC Fotodiode Pada Saat Tidak Terhalang

4.3 Pengukuran dan Pengujian Pergerakan Motor DC pada Buka-Tutup Valve Tangki Pakan

Tujuan dari pengujian pergerakan motor DC pada buka-tutup *valve* tangki pakan adalah untuk mengetahui selisih derajat sudut awal dan sudut akhir ketika sistem aktif dan persentase kesalahannya.

Untuk melakukan pengujian ini, dilakukan pemrograman melalui Arduino untuk membaca nilai ADC potensiometer yang dikopel motor DC

melalui komunikasi serial dengan komputer. Kemudian untuk data tegangan diukur tegangan pada pin *wiper* dan pin *ground* potensiometer saat rangkaian diaktifkan.

Berdasarkan teori, analog input arduino memiliki kisaran tegangan antara 0 sampai dengan 5V untuk kisaran nilai ADC 0 sampai dengan 1023. Sehingga nilai tegangan untuk setiap ADC adalah sebagai berikut :

$$5V : 1024 = 0,0048876 \text{ V}$$

Kemudian apabila potensiometer 10K memiliki nilai ADC antara 0 sampai dengan 1023 pada saat posisi derajat porosnya 0 sampai dengan 360 derajat, maka nilai ADC setiap derajatnya adalah :

$$1023 : 360 = 2,8417$$

Sehingga nilai tegangan untuk setiap derajat gerakan potensiometer adalah :

$$2,8417 \times 0,0048876 \text{ V} = 0,01388909292 \text{ V}$$

Hasil diatas merupakan teori perhitungan, untuk nilai asli yang terbaca dari potensiometer ketika diukur menggunakan AVOMeter pada saat nilai ADC nya dibaca melalui serial monitor arduino dapat diamati pada tabel 4.7 berikut.

Tabel 4.7 Data Pengukuran Sensor Posisi Potensiometer 1

No.	Sudut <i>Valve</i> Tangki Pakan	Tegangan Potensiometer (Volt)	Nilai ADC Potesiometer	Kondisi <i>Valve</i>
1	0	1,163	236	Tertutup
2	45	1,614	329	Terbuka Separuh
3	90	2,081	424	Terbuka Penuh

Selisih ADC potensiometer untuk setiap 45 derajat perubahan sudut antara sudut 0 derajat sampai dengan 90 derajat adalah sebesar 94.

4.4 Pengukuran dan Pengujian Pergerakan Motor DC Pengatur Posisi *Blower*

Tujuan dari pengujian pergerakan motor DC pengatur posisi *Blower* adalah untuk mengetahui selisih derajat sudut awal dan sudut akhir ketika sistem aktif dan persentase kesalahannya.

Untuk melakukan pengujian ini, dilakukan pemrograman melalui Arduino untuk membaca nilai ADC potensiometer yang dikopel motor DC melalui komunikasi serial dengan komputer. Data tegangan diambil dengan mengukur tegangan pada pin *wiper* dan pin *ground* potensiometer saat rangkaian diaktifkan.

Tabel 4.8 Data Pengukuran Sensor Posisi Potensiometer 2

No.	Sudut Mulut <i>Blower</i>	Tegangan Hasil Pengukuran (Volt)	Nilai ADC Potensiometer
1.	0	3,000	613
2.	10	3,148	642
3.	15	3,244	662
4.	30	3,468	708
5.	45	3,720	758
6.	60	3,943	805

Rata rata selisih ADC untuk setiap 15 derajat perubahan sudut antara sudut 0 sampai 60 derajat adalah sebesar 48.

4.5 Pengukuran dan Pengujian Keluaran Sensor Hujan

Tujuan dari pengujian sensor hujan ini adalah untuk menentukan nilai ADC yang tepat sebagai tolok ukur kondisi saat ini terjadi hujan atau tidak. Sensor hujan dijadikan syarat bagi sistem untuk melakukan proses pemberian pakan atau tidak, sehingga ketepatannya cukup berpengaruh pada kerja sistem.

Metode yang dilakukan untuk melakukan pengukuran ini adalah dengan menyambung sensor ke pin analog arduino serta memasang penghantar pada pin input sensor hujan. Lantas penghantar dikenakan air sehingga basah dan kedua polaritasnya tersambung. Selanjutnya tegangan yang keluar menuju input analog arduino akan diukur dengan AVOMeter untuk membandingkan nilai ADC yang terbaca pada serial monitor dengan nilai tegangan yang ada.

Tabel 4.4 Data Pengukuran Sensor Hujan

No	Sensor Hujan	Tegangan (V)
1	525	4.87
2	520	4.87
3	519	4.87
4	518	4.87
5	517	4.87
6	515	4.87
7	512	4.87
8	509	4.87
9	508	4.87
10	500	4.50
11	498	4.50
12	496	4.50
13	498	4.40
14	497	4.40
15	496	4.40

4.6 Pengujian Jarak Lontar Pakan dengan Blower

Tujuan dari pengujian jarak lontar ini adalah mengetahui seberapa jauh pakan dapat tersebar apabila sudut mulut blower diubah – ubah. Hasil yang didapat nantinya akan dijadikan referensi untuk menentukan nilai derajat yang sesuai untuk digunakan pada mekanisme alat. Untuk melakukan pengujian ini, *blower* diubah ubah posisi derajatnya dan dicatu dengan tegangan DC 24 V. Kemudian pakan diberikan dengan takaran tertentu melalui mulut blower dan jaraknya diukur menggunakan meteran rol. Hasil yang didapat dapat dilihat pada tabel berikut :

Tabel 4.5 Data Jarak Lontar *Blower*

No	Sudut	Jarak Konsentrasi
1	0	1 meter
2	15	1,2 meter
3	30	1,4 meter

-----Halaman ini sengaja dikosongkan-----

BAB V

PENUTUP

Dari proses perancangan alat dan pengambilan data yang kami lakukan, terdapat beberapa kesimpulan yang dapat kami ambil. Selain itu terdapat juga beberapa saran untuk penelitian selanjutnya agar dapat diperbaiki dan dikembangkan sehingga dapat lebih baik dan lebih bermanfaat.

5.1 Kesimpulan

1. Dari pengujian sensor fotodioda, pembacaan ADC fotodioda pada arduino bernilai rendah ketika fotodioda tidak terhalang dengan rata-rata nilai tegangan sebesar 0,122 volt. Sedangkan pada saat fotodioda terhalang pakan ikan, ADC bernilai tinggi dengan rata-rata nilai tegangan 4,82 volt. Hasil ADC ini dapat digunakan sebagai parameter untuk mengatur nyala lampu indikator *level* tangki pakan.
2. Pada pengujian sensor posisi menggunakan potensiometer, pembacaan ADC potensiometer sangat mempengaruhi akurasi pergerakan posisi sudut motor DC, hal ini dikarenakan program sudah diatur untuk memenuhi nilai ADC tertentu yang dibaca dari potensiometer.
3. Selisih ADC potensiometer pada motor penggerak *blower* adalah sebesar 48 untuk setiap 15 derajat perubahan sudut dengan selisih tegangan rata-rata 0,244 volt untuk sudut 0 sampai 60 derajat. Sedangkan untuk potensiometer pada motor penggerak *valve* tangki selisih perubahan ADC adalah sebesar 94 untuk setiap 45 derajat perubahan sudut dengan selisih tegangan rata-rata sekitar 0,46 volt untuk sudut 0 sampai dengan 90 derajat.
4. Jarak lontar yang dicapai pada saat pakan yang jatuh ditiup oleh *blower* memiliki selisih sekitar 0,2 meter untuk setiap 15 derajat kenaikan sudut blower untuk selisih sudut 0, 15, dan 30 derajat. Sedangkan jarak maksimalnya tidak menentu.
5. Dari hasil pengujian sensor hujan, ADC pin sensor hujan bernilai tinggi ketika belum terkena air sama sekali. Kemudian berangsur-angsur menurun ketika konduktornya terkena air. Laju penurunan ADC yang terbaca tergantung pada intensitas air yang mengenai permukaan konduktor sensor hujan.

6. Arus balik yang terjadi ketika motor DC aktif dapat mempengaruhi kinerja rangkaian karena dapat mengakibatkan kontroler mengalami *reset*. Rangkaian dioda pengaman dapat mengatasi timbulnya arus balik ini, akan tetapi apabila arus balik terlalu besar dioda pengaman terlampaui kapasitasnya sehingga arus balik kembali mengakibatkan rangkaian *reset*.

5.2 Saran

Untuk penelitian selanjutnya, sebaiknya digunakan kontroler dan rangkaian *power supply* dengan kondisi yang benar-benar baik karena hal ini berpengaruh pada kualitas data yang diakuisisi oleh sensor. Kemudian ada baiknya gerakan mekanis motor ditambah secara horizontal agar persebaran pakan lebih merata dan efisien. Untuk antarmuka pengguna, akan lebih baik kalau terdapat *monitoring* secara *wireless* melalui *smartphone* android maupun SMS.

DAFTAR PUSTAKA

- [1] Douglaz Krantz, “*Flyback Diode*”, <http://www.douglaskrantz.com/ElecFlybackDiode.html>, diakses tanggal 19 Juli 2017 pukul 11.10
- [2] Emmanuel Ozigbo dan Anyadike Chinenye, “*Development of an Automatic Fish Feeder*”, Paper, Department of Agricultural and Bioresources Engineering, University of Nigeria Nsukka, 2013.
- [3] Evi Siska Safarina & Hendry Erwantono, “*Rancang Bangun Lengan Robot 3 DOF Berbasis ATMEGA 328 untuk Memindahkan Biskuit*”, Tugas Akhir, Program Studi D III Teknik Elektro, Institut Teknologi Sepuluh Nopember, 2016.
- [4] Miraditya P, “*Rancang Bangun Alat Pemesanan Menu Makanan Otomatis Berbasis Mikrokontroler dengan Komunikasi TCP/IP*”, Tugas Akhir, Jurusan Sistem Komputer, Sekolah Tinggi Manajemen Informatika dan Teknik Komputer Surabaya, 2014.
- [5] Muhammad Yusuf, “*Prototipe Sensor Parkir Mobil Berbasis Mikrokontroler AT89S51*”, Tugas Akhir, Program Diploma III Ilmu Komputer, Universitas Sebelas Maret Surakarta, 2009
- [6] Nuning Setyaningrum, “*Teknik Pembesaran Ikan Gurami*”, <http://bio.unsoed.ac.id/sites/default/files/Teknik%20Pembesaran%20Ikan%20Gurami-0.pdf>, diakses pada tanggal 19 Juli 2017 pukul 11.51
- [7] Patrick Henry G Baniqued, Martin Joseph de Castro, Chael Triston T. Luzano, *Microcontroller Based Fish Feeder*, Design Report, Bachelor of Science in Computer Engineering, Mapua Institute of Technology, 2009
- [8] _____, *Photodiode*, <https://en.wikipedia.org/wiki/Photodiode>, Diakses pada tanggal 19 juli 2017 pukul 11.48.
- [9] _____, *Prinsip Dasar Cara Kerja Motor DC Sederhana*, <https://ilmuteknik.com/mesin/prinsip-dasar-cara-kerja-motor-dc-sederhana.html/>, Diakses pada tanggal 19 juli 2017 pukul 12.42

-----Halaman ini sengaja dikosongkan-----

LAMPIRAN

A. Listing Program

```
#include <Wire.h>
#include <OneWire.h>          //library sensor temperatur onewire
#include <DallasTemperature.h> //library sensor dallas temperatur
#include <DS3231.h>           //Memanggil RTC3231 Library
#include <Keypad.h>           //library Keypad
#include <LiquidCrystal_I2C.h> //library LCD I2C

int posisiValve;
int valveDulu;
int posisi;
int posisiSebelum;
const int blowerPin = 51;
const int IN1 = 33;
const int IN2 = 35;
const int IN3 = 37;
const int IN4 = 39;
const int potensioPin = A8;
const int valvePin = A9;
const int indicator_A = 45;
const int indicator_B = 43;
const int indicator_C = 41;
const int sensor_A = A12;
const int sensor_B = A11;
const int sensor_C = A10;
int ADC_A;
int ADC_B;
int ADC_C;
unsigned long dulu = 0;
const long selisih = 250;
int switching = LOW;

#define SensorPin A1          // ke Arduino Analog Input 8
#define Offset 0.00
#define samplingInterval 20
#define printInterval 1000
#define ArrayLenth 40
```

```

int pHArray[ArrayLenth];
int pHArrayIndex=0;

int jm;
int mt;
int jm2;
int mt2;
int aktifkan;

#define oneWireBus 53           //Inialisasi pin sensor temperatur
OneWire oneWire(oneWireBus);    //jenis sensor onewire
DallasTemperature sensors(&oneWire); //sensor onewire dallas temp
float Temp1 = 0;                //data temp 1
float Temp2 = 1;                //data temp 2
float Temp3 = 2;                //data temp 3
float Temp4 = 3;                //data temp 4
float Temp5 = 4;                //data temp 5
float Temp;                     //data all temp
int a,b=0;

const int pinHujan = A0; //Inialisasi pin sensor hujan
int dataHujan;           //Deklarasi variabel data pembacaan sensor hujan
const int pompa = 49;    //Inialisasi pin pompa

DS3231 rtc(SDA, SCL);
Time t;                 //pencacah string time()
int jam = 0, mnt = 0, dtk = 0, tgl = 0, bln = 0; //Mengambil waktu jam
int thn = 0;
uint8_t alarmMode=0;
String masukJam="";     //data jam pakan dari keypad
String masukMenit="";   //data menit pakan dari keypad
String masukJamm="";    //data jam pengairan dari keypad
String masukMenitt="";  //data menit pengairan dari keypad
String lamaJam;         //data lama pengairan dari keypad
String lamaMenit;

bool tampilanAwal=false;
bool setWaktu = false;
bool masukJadwal = false;

```

```

bool setJadwal = false;
bool jadwalJam = false;
bool jadwalMenit = false;
bool jadwalPump = false;
bool jadwalPumpp = false;

char keypress;
const byte ROWS = 4;    //jumlah baris keypad
const byte COLS = 3;    //jumlah kolom keypad
char keys[ROWS][COLS] = {
  {'1','2','3'},
  {'4','5','6'},
  {'7','8','9'},
  {'*','0','#'}};
byte rowPins[ROWS] = {8,7,6,5}; //pin" baris untuk keypad
byte colPins[COLS] = {4,3,2};   //pin" kolom untuk keypad
Keypad myKeypad( makeKeymap(keys), rowPins, colPins, ROWS,
COLS);

```

```

LiquidCrystal_I2C lcd(0x3F ,2,1,0,4,5,6,7,3, POSITIVE); //0x3F =
alamat i2C LCD

```

```

void setup() {
  rtc.begin();           //memulai koneksi i2c dengan RTC
  Serial.begin(9600);
  lcd.begin(20,4);       //memulai koneksi LCD
  sensors.begin();       //memulai koneksi sensor"
  pinMode(pinHujan, INPUT); //Inialisasi input pin hujan
  pinMode(pompa, OUTPUT); //Inialisasi output pompa
  // pinMode(blower, OUTPUT);
  pinMode(indicator_A, OUTPUT);
  pinMode(indicator_B, OUTPUT);
  pinMode(indicator_C, OUTPUT);
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);
  pinMode(blowerPin, OUTPUT);
  Serial.begin(9600);
}

```

```

digitalWrite(IN1, HIGH);
digitalWrite(IN2, HIGH);
digitalWrite(IN3, HIGH);
digitalWrite(IN4, HIGH);
}
unsigned long e,f=0;
int x=1;
unsigned long c,d=0;
int y=1;

void loop(){
  if(tampilanAwal){ //tampilan awal LCD
    myRTC();        //memanggil void myRTC
    indikator();
    pHdanTemp(Temp); //memanggil void pH dan Temperatur
    Serial.print(masukJam); Serial.print(" ");Serial.println(jm);
    lcd.setCursor(0,2);
    lcd.print("Jadwal Pkn=");lcd.setCursor(12,2);lcd.print(masukJam);
    lcd.setCursor(14,2);lcd.print(':');lcd.print(masukMenit);
    lcd.setCursor(0,3);
    lcd.print("Jdwl Air=");lcd.setCursor(10,3);lcd.print(masukJamm);
    lcd.setCursor(12,3);lcd.print(':');lcd.print(masukMenitt);
    lcd.setCursor(16,3);lcd.print(lamaJam);lcd.print(lamaMenit);
    setWaktu = true; //menuju setWaktu
    //alarmAir();
    alarmPakan();
  }
  if(!setWaktu){ //jika setwaktu salah, maka muncul tampilan
awal
    setWaktu = false;
    tampilanAwal = true;
  }
  if(setWaktu == true){ //jika setwaktu benar
    keypress = myKeypad.getKey();
    if(keypress == '*'){ //ketika tekan (*) maka muncul tampilan menu
      tampilanAwal = false;
      lcd.clear();
      lcd.setCursor(0,0);lcd.print("Menu Utama");
    }
  }
}

```

```

        lcd.setCursor(0,1);lcd.print("1-Jadwal Pakan");          //1 untuk
jadwal pakan
        lcd.setCursor(0,2);lcd.print("2-Jadwal Air");          //2 untuk jadwal
pengairan
        lcd.setCursor(0,3);lcd.print("3-Lama Waktu Pompa"); //3 untuk
lama pengairan
        masukJadwal = true;
    }
    else if(keypress == '#'){ //ketika tekan # maka kembali ke tampilan
awal
        lcd.clear();
        tampilanAwal = true;
    }}
    if(masukJadwal ==true){ //jika masukJadwal benar
        jadwalPakan();          //memanggil void jadwalPakan
        jadwalAir();          //memanggil void jadwalAir
        lamaPompa();          //memanggil void jadwalAir
    }
}

void indikator(){
    ADC_A = analogRead(sensor_A);
    ADC_B = analogRead(sensor_B);
    ADC_C = analogRead(sensor_C);
    if(ADC_A > 200 && ADC_B > 200 && ADC_C > 200){
        digitalWrite(indicator_A, LOW);
        digitalWrite(indicator_B, LOW);
        digitalWrite(indicator_C, LOW);}
    if(ADC_A < 200 && ADC_B > 200 && ADC_C > 200){
        digitalWrite(indicator_A, HIGH);
        digitalWrite(indicator_B, LOW);
        digitalWrite(indicator_C, LOW);}
    if(ADC_A < 200 && ADC_B < 200 && ADC_C > 200){
        digitalWrite(indicator_A, HIGH);
        digitalWrite(indicator_B, HIGH);
        digitalWrite(indicator_C, LOW);}
    if(ADC_A < 200 && ADC_B < 200 && ADC_C < 200){
        unsigned long sekarang = millis();
        digitalWrite(indicator_A, HIGH);
        digitalWrite(indicator_B, HIGH);

```

```

    if(sekarang - dulu >= selisih){
        dulu = sekarang;
    }
    if(switching == LOW){
        switching = HIGH;
    }
    else{switching = LOW;}
    digitalWrite(indicator_C, switching);
}
}

void myRTC(){
    t = rtc.getTime();
    jam = t.hour; mnt = t.min; dtk = t.sec;
    tgl = t.date; bln = t.mon; thn = t.year;
    lcd.setCursor(0,0);
    lcd.print(rtc.getDOWStr(1));lcd.print(',');
    tampil(tgl);lcd.print('/');tampil(bln);
    lcd.setCursor(11,0);
    tampil(jam);lcd.print(':');tampil(mnt);lcd.print(':');tampil(dtk);
    alarmPakan();
    //alarmAir();
}

void tampil(int digit){
    if(digit >= 0 && digit < 10){
        lcd.print('0');
        lcd.print(digit);
    }
}

void pHdanTemp(float Temp){
    //program sensor pH
    static unsigned long samplingTime = millis();
    static unsigned long printTime = millis();
    static float pHValue,voltage;
    if(millis()-samplingTime > samplingInterval){
        pHArray[pHArrayIndex++]=analogRead(SensorPin);
        if(pHArrayIndex==ArrayLenth)pHArrayIndex=0;
        voltage = avergearray(pHArray, ArrayLenth)*5.0/1024;
        pHValue = 3.5*voltage+Offset;
        samplingTime=millis();
    }
    if(millis()-printTime > printInterval){ //Setiap 1000 ms, print nilai
        lcd.setCursor(0,1);
        lcd.print("pH=");
    }
}

```



```

    lcd.print(pHValue);
    Serial.print("Teg:");
    Serial.print(voltage);Serial.print(" ");
    printTime=millis();}
    //program sensor temperatur
    sensors.requestTemperatures();
    Temp1 = sensors.getTempCByIndex(0);
    Temp2 = sensors.getTempCByIndex(1);
    Temp3 = sensors.getTempCByIndex(2);
    Temp4 = sensors.getTempCByIndex(3);
    Temp5 = sensors.getTempCByIndex(4);
    Temp = (Temp1+Temp2+Temp3+Temp4+Temp5)/5;
    lcd.setCursor(9,1);
    lcd.print("T=");
    lcd.print(Temp);
    if(pHValue >= 6.5 && pHValue <= 7.15 && (Temp >= 24 &&
Temp <= 30)){digitalWrite(pompa,LOW);}
    else{ digitalWrite(pompa,HIGH);}
    sensorHujan();
}
double avergearray(int* arr, int number){
    int i;
    int max,min;
    double avg;
    long amount=0;
    if(number<=0){
        Serial.println("Error number for the array to avraging!/n");
        return 0;}
    if(number<5){ //<5, menghitung langsung statistik
        for(i=0;i<number;i++){
            amount+=arr[i];}
        avg = amount/number;
        return avg;
    }else{
        if(arr[0]<arr[1]){
            min = arr[0];max=arr[1];
        }else{
            min=arr[1];max=arr[0];}
        for(i=2;i<number;i++){

```

```

    if(arr[i]<min){
        amount+=min;    //arr<min
        min=arr[i];
    }else{
        if(arr[i]>max){
            amount+=max;    //arr>max
            max=arr[i];
        }else{
            amount+=arr[i]; } //min<=arr<=max
        } }
    avg = (double)amount/(number-2);}
    return avg;
}

void sensorHujan(){
    dataHujan = analogRead(pinHujan);
    if(dataHujan <= 450){
        digitalWrite(pompa,LOW);}
    }

void jadwalPakan(){
    keypress = myKeypad.getKey();
    if(keypress == '1'){
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("Jadwal Untuk Pakan");
        lcd.setCursor(2,1);lcd.print(':');
        masukJam="";
        jadwalJam = true;
        int j=0;
        while(jadwalJam){
            keypress = myKeypad.getKey();
            if(keypress != NO_KEY){
                if(keypress == '1' || keypress == '2' || keypress == '3' || keypress
== '4' ||
                    keypress == '5' || keypress == '6' || keypress == '7' || keypress
== '8' ||
                        keypress == '9' || keypress == '0'){
                            masukJam += keypress;
                            lcd.setCursor(j,1);
                            lcd.print(keypress);

```

```

        j++;
    }}
    if(j >= 2){
        jadwalJam = false;
        masukMenit="";
        jadwalMenit = true;
        int m=3;
        while(jadwalMenit){
            keypress = myKeypad.getKey();
            if(keypress != NO_KEY){
                if(keypress == '1' || keypress == '2' || keypress == '3' || keypress
== '4' ||
                keypress == '5' || keypress == '6' || keypress == '7' || keypress
== '8' ||
                keypress == '9' || keypress == '0'){
                    masukMenit += keypress;
                    lcd.setCursor(m,1);
                    lcd.print(keypress);
                    m++;
                }
            }
            keypress = myKeypad.getKey();
            if(m >= 6 || keypress == '*'){
                jadwalMenit = false;
                lcd.clear();
                lcd.setCursor (0,0);
                lcd.print("Pemberi Pakan Aktif");
                lcd.setCursor (0,1);
                lcd.print("pada : ");
                lcd.print(masukJam); lcd.print(':');
                lcd.print(masukMenit);
                alarmMode=1;
                delay(3000);
                lcd.clear();
                tampilanAwal = true;
            }
        }
    }
}

void jadwalAir(){
    keypress = myKeypad.getKey();

```

```

if(keypress == '2'){
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Jadwal Pengairan");
    lcd.setCursor(2,1);lcd.print(':');
    masukJamm="";
    jadwalJam = true;
    int jj=0;
    while(jadwalJam){
        keypress = myKeypad.getKey();
        if(keypress != NO_KEY){
            if(keypress == '1' || keypress == '2' || keypress == '3' || keypress
== '4' ||
            keypress == '5' || keypress == '6' || keypress == '7' || keypress
== '8' ||
            keypress == '9' || keypress == '0'){
                masukJamm += keypress;
                lcd.setCursor(jj,1);
                lcd.print(keypress);
                jj++;
            }
        }
        if(jj >= 2){
            jadwalJam = false;
            masukMenitt="";
            jadwalMenit = true;
            int mm=3;
            while(jadwalMenit){
                keypress = myKeypad.getKey();
                if(keypress != NO_KEY){
                    if(keypress == '1' || keypress == '2' || keypress == '3' || keypress
== '4' ||
                    keypress == '5' || keypress == '6' || keypress == '7' || keypress
== '8' ||
                    keypress == '9' || keypress == '0'){
                        masukMenitt += keypress;
                        lcd.setCursor(mm,1);
                        lcd.print(keypress);
                        mm++;
                    }
                }
            }
        }
    }
}

```

```

    keypress = myKeypad.getKey();
    if(mm >= 6 || keypress == '*'){
        jadwalMenit = false;
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("Sirkulasi Air Aktif");
        lcd.setCursor(0,1);
        lcd.print("pada : ");
        lcd.print(masukJamm); lcd.print(':');
        lcd.print(masukMenitt);
        alarmMode=1;
        delay(3000);
        lcd.clear();
        tampilanAwal=true;
    }}}}

}}
void lamaPompa(){
    keypress = myKeypad.getKey();
    if(keypress == '3'){
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("Lama Pengairan");
        lcd.setCursor(3,1);lcd.print("jam");
        lcd.setCursor(3,2);lcd.print("menit");
        lamaJam="";
        jadwalPump = true;
        int tt=0;
        while(jadwalPump){
            keypress = myKeypad.getKey();
            if(keypress != NO_KEY){
                if(keypress == '1' || keypress == '2' || keypress == '3' || keypress
== '4' ||
                    keypress == '5' || keypress == '6' || keypress == '7' || keypress
== '8' ||
                        keypress == '9' || keypress == '0'){
                    lamaJam += keypress;
                    lcd.setCursor(tt,1);
                    lcd.print(keypress);

```

```

        tt++;
    }}
    if(tt >= 2 || keypress == '*'){
        jadwalPump = false;
        lamaMenit="";
        jadwalPumpp = true;
        int ss=0;
        while(jadwalPumpp){
            keypress = myKeypad.getKey();
            if(keypress != NO_KEY){
                if(keypress == '1' || keypress == '2' || keypress == '3' || keypress
== '4' ||
                keypress == '5' || keypress == '6' || keypress == '7' || keypress
== '8' ||
                keypress == '9' || keypress == '0'){
                    lamaMenit += keypress;
                    lcd.setCursor(ss,2);
                    lcd.print(keypress);
                    ss++;
                }}
            keypress = myKeypad.getKey();
            if(ss >= 2 || keypress == '*'){
                jadwalPumpp = false;
                lcd.clear();
                lcd.setCursor(0,0); lcd.print("Sirkulasi Air Aktif");
                lcd.setCursor(0,1); lcd.print("selama :");
                lcd.setCursor(0,2);   lcd.print(lamaJam);   lcd.setCursor(3,2);
lcd.print("jam");
                lcd.setCursor(7,2);   lcd.print(lamaMenit);   lcd.setCursor(10,2);
lcd.print("menit");
                alarmMode=1;
                delay(3000);
                lcd.clear();
                tampilanAwal=true;
            }}}
        }

    }}

```

```

void alarmPakan(){
    konversi();
    if (jam == jm && mnt == mt){
        tumhiho();
    }
}

```

```

void konversi(){
    if(masukJam == "00"){
        jm = 0;
        jm2 = 0;
    }
    else if(masukJam == "01"){
        jm = 1;
        jm2 = 1;
    }
    else if(masukJam == "02"){
        jm = 2;
        jm2 = 2;
    }
    else if(masukJam == "03"){
        jm = 3;
        jm2 = 3;
    }
    else if(masukJam == "04"){
        jm = 4;
        jm2 = 4;
    }
    else if(masukJam == "05"){
        jm = 5;
        jm2 = 5;
    }
    else if(masukJam == "06"){
        jm = 6;
        jm2 = 6;
    }
    else if(masukJam == "07"){
        jm = 7;
        jm2 = 7;
    }
}

```

```

    }
    else if(masukJam == "08"){
        jm = 8;
        jm2 = 8;
    }
    else if(masukJam == "09"){
        jm = 9;
        jm2 = 9;
    }
    else if(masukJam == "10"){
        jm = 10;
        jm2 = 10;
    }

    else if(masukJam == "11"){
        jm = 11;
        jm2 = 11;
    }
    else if(masukJam == "12"){
        jm = 12;
        jm2 = 12;
    }
    else if(masukJam == "13"){
        jm = 13;
        jm2 = 13;
    }
    else if(masukJam == "14"){
        jm = 14;
        jm2 = 14;
    }
    else if(masukJam == "15"){
        jm = 15;
        jm2 = 15;
    }
    else if(masukJam == "16"){
        jm = 16;
        jm2 = 16;
    }
    else if(masukJam == "17"){

```



```

jm = 17;
jm2 = 17;
}
else if(masukJam == "18"){
jm = 18;
jm2 = 18;
}
else if(masukJam == "19"){
jm = 19;
jm2 = 19;
}
else if(masukJam == "20"){
jm = 20;
jm2 = 20;
}
else if(masukJam == "21"){
jm = 21;
jm2 = 21;
}
else if(masukJam == "22"){
jm = 22;
jm2 = 22;
}
else if(masukJam == "23"){
jm = 23;
jm2 = 23;
}
else if(masukJam == "24"){
jm = 24;
jm2 = 24;
}

if(masukMenit == "00"){
mt = 0;
mt2 = 0;
}
else if(masukMenit == "01"){
mt = 1;
mt2 = 1;
}

```

```

    }
    else if(masukMenit == "02"){
        mt = 2;
        mt2 = 2;
    }
    else if(masukMenit == "03"){
        mt = 3;
        mt2 = 3;
    }
    else if(masukMenit == "04"){
        mt = 4;
        mt2 = 4;
    }
    else if(masukMenit == "05"){
        mt = 5;
        mt2 = 5;
    }
    else if(masukMenit == "06"){
        mt = 6;
        mt2 = 6;
    }
    else if(masukMenit == "07"){
        mt = 7;
        mt2 = 7;
    }
    else if(masukMenit == "08"){
        mt = 8;
        mt2 = 8;
    }
    else if(masukMenit == "09"){
        mt = 9;
        mt2 = 9;
    }
    else if(masukMenit == "10"){
        mt = 10;
        mt2 = 10;
    }
    else if(masukMenit == "11"){
        mt = 11;

```

```

    mt2 = 11;
}
    else if(masukMenit == "12"){
        mt = 12;
        mt2 = 12;
    }
    else if(masukMenit == "13"){
        mt = 13;
        mt2 = 14;
    }
    else if(masukMenit == "14"){
        mt = 14;
        mt2 = 14;
    }
    else if(masukMenit == "15"){
        mt = 15;
        mt2 = 15;
    }
    else if(masukMenit == "16"){
        mt = 16;
        mt2 = 16;
    }
    else if(masukMenit == "17"){
        mt = 17;
        mt2 = 17;
    }
    else if(masukMenit == "18"){
        mt = 18;
        mt2 = 18;
    }
    else if(masukMenit == "19"){
        mt = 19;
        mt2 = 19;
    }
    else if(masukMenit == "20"){
        mt = 20;
        mt2 = 20;
    }
    else if(masukMenit == "21"){

```

```

    mt = 21;
    mt2 = 21;
}
    else if(masukMenit == "22"){
    mt = 22;
    mt2 = 22;
}
else if(masukMenit == "23"){
    mt = 23;
    mt2 = 23;
}
else if(masukMenit == "24"){
    mt = 24;
    mt2 = 24;
}
else if(masukMenit == "25"){
    mt = 25;
    mt2 = 25;
}
    else if(masukMenit == "26"){
    mt = 26;
    mt2 = 26;
}
else if(masukMenit == "27"){
    mt = 27;
    mt2 = 27;
}
    else if(masukMenit == "28"){
    mt = 28;
    mt2 = 28;
}
else if(masukMenit == "29"){
    mt = 29;
    mt2 = 29;
}
else if(masukMenit == "30"){
    mt = 30;
    mt2 = 30;
}

```

```
else if(masukMenit == "31"){
    mt = 31;
    mt2 = 31;
}
    else if(masukMenit == "32"){
        mt = 32;
        mt2 = 32;
    }
else if(masukMenit == "33"){
    mt = 33;
    mt2 = 33;
}
    else if(masukMenit == "34"){
        mt = 34;
        mt2 = 34;
    }
else if(masukMenit == "35"){
    mt = 35;
    mt2 = 35;
}
    else if(masukMenit == "36"){
        mt = 36;
        mt2 = 36;
    }
else if(masukMenit == "37"){
    mt = 37;
    mt2 = 37;
}
    else if(masukMenit == "38"){
        mt = 38;
        mt2 = 38;
    }
    else if(masukMenit == "39"){
        mt = 39;
        mt2 = 39;
    }
else if(masukMenit == "40"){
    mt = 40;
    mt2 = 40;
```

```

}
    else if(masukMenit == "41"){
        mt = 41;
        mt2 = 41;
    }
    else if(masukMenit == "42"){
        mt = 42;
        mt2 = 42;
    }
    else if(masukMenit == "43"){
        mt = 43;
        mt2 = 43;
    }
    else if(masukMenit == "44"){
        mt = 44;
        mt2 = 44;
    }
    else if(masukMenit == "45"){
        mt = 45;
        mt2 = 45;
    }
    else if(masukMenit == "46"){
        mt = 46;
        mt2 = 46;
    }
    else if(masukMenit == "47"){
        mt = 47;
        mt2 = 47;
    }
    else if(masukMenit == "48"){
        mt = 48;
        mt2 = 48;
    }
    else if(masukMenit == "49"){
        mt = 49;
        mt2 = 49;
    }
    else if(masukMenit == "50"){
        mt = 50;
    }

```

```

    mt2 = 50;
}
    else if(masukMenit == "51"){
    mt = 51;
    mt2 = 51;
}
    else if(masukMenit == "52"){
    mt = 52;
    mt2 = 52;
}
    else if(masukMenit == "53"){
    mt = 53;
    mt2 = 53;
}
    else if(masukMenit == "54"){
    mt = 54;
    mt2 = 54;
}
    else if(masukMenit == "55"){
    mt = 55;
    mt2 = 55;
}
    else if(masukMenit == "56"){
    mt = 56;
    mt2 = 56;
}
    else if(masukMenit == "57"){
    mt = 57;
    mt2 = 57;
}
    else if(masukMenit == "58"){
    mt = 58;
    mt2 = 58;
}
    else if(masukMenit == "59"){
    mt = 59;
    mt2 = 59;
}
}

```

```

void tumhiho(){
if(x = 1){
    digitalWrite(blowerPin, HIGH);
    e=millis();
    f=millis();
    while((f-e) <= 5000){
        f=millis();
    }
    pakan();
    //Start motor
    posisiSebelum = analogRead(potensioPin);
    posisi = analogRead(potensioPin);
    while((posisi-posisiSebelum) <= 75){
        putarKanan();
        posisi = analogRead(potensioPin);
    }
    Serial.println("Stage1");
    berhenti();
    pakan();
    //Delay 2 detik
    e=millis();
    f=millis();
    while((f-e) <= 2000){
        f=millis();
    }
    while((posisi-posisiSebelum) <= 150){
        putarKanan();
        posisi = analogRead(potensioPin);
    }
    Serial.println("Stage2");
    berhenti();
    y=1;
    pakan();
    //Delay 2 detik
    e=millis();
    f=millis();
    while((f-e) <= 2000){
        f=millis();

```



```

    }
    while((posisi-posisiSebelum) >= 0){
        putarKiri();
        posisi = analogRead(potensioPin);
    }
    Serial.println("Stage3");
    berhenti();
    x=x+1;
    digitalWrite(blowerPin, LOW);
}
}
void berhenti(){
    digitalWrite(IN3, HIGH);
    digitalWrite(IN4, HIGH);
}
void putarKanan(){
    digitalWrite(IN3, HIGH);
    digitalWrite(IN4, LOW);
}
void putarKiri(){
    digitalWrite(IN3, LOW);
    digitalWrite(IN4, HIGH);
}

void valveKiri(){
    digitalWrite(IN1, HIGH);
    digitalWrite(IN2, LOW);
}

void valveKanan(){
    digitalWrite(IN1, LOW);
    digitalWrite(IN2, HIGH);
}

void valveStop(){
    digitalWrite(IN1, HIGH);
    digitalWrite(IN2, HIGH);
}

```

```

void pakan(){
  if(y==1){
    valveDulu = analogRead(valvePin);
    posisiValve = analogRead(valvePin);
    while((posisiValve-valveDulu) <= 100){
      valveKanan();
      posisiValve = analogRead(valvePin);
    }
    Serial.println("step1");
    valveStop();
    //delay
    c=millis();
    d=millis();
    while((d-c) <= 1000){
      d=millis();
    }
    while((posisiValve-valveDulu) >= -50){
      valveKiri();
      posisiValve = analogRead(valvePin);
    }
    Serial.println("step2");
    valveStop();
    y=0;
  }
}

```

B. Dokumentasi Alat



Gambar 7.1 Fisik Alat Pemberi Pakan dari Samping



Gambar 7.2 Fisik Alat Pemberi Pakan dari Belakang



Gambar 7.3 Fisik Alat Pemberi Pakan dari Depan



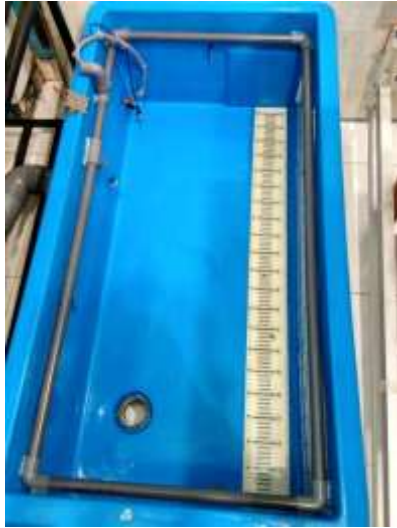
Gambar 7.4 Pemasangan Blower pada Motor Pengatur Sudut



Gambar 7.5 *Box* Kontrol dengan Antarmuka LCD dan Keypad



Gambar 7.6 Rangkaian Dalam *Box* Kontrol



Gambar 7.7 Kolam untuk Simulasi Pemberian Pakan dan Pengaturan Sirkulasi Air



Gambar 7.8 Pemasangan Modul Sensor Hujan

C. Datasheet

B.1 DS3231 Real Time Clock

DS3231

Extremely Accurate I²C-Integrated
RTC/TCXO/Crystal

Absolute Maximum Ratings

Voltage Range on Any Pin Relative to Ground.....-0.3V to +6.0V
Junction-to-Ambient Thermal Resistance (θ_{JA}) (Note 1) 73°C/W
Junction-to-Case Thermal Resistance (θ_{JC}) (Note 1) 23°C/W
Operating Temperature Range
DS3231S0°C to +70°C
DS3231SN-40°C to +85°C

Junction Temperature+125°C
Storage Temperature Range-40°C to +85°C
Lead Temperature (soldering, 10s)+260°C
Soldering Temperature (reflow, 2 times max)+260°C
(see the Handling, PCB Layout, and Assembly section)

Note 1: Package thermal resistances were obtained using the method described in JEDEC specification JESD51-7, using a four-layer board. For detailed information on package thermal considerations, refer to www.maximintegrated.com/thermal-tutorial.

Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Recommended Operating Conditions

($T_A = T_{MIN}$ to T_{MAX} , unless otherwise noted.) (Notes 2, 3)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Supply Voltage	V_{CC}		2.3	3.3	5.5	V
	V_{BAT}		2.3	3.0	5.5	V
Logic 1 Input SDA, SCL	V_{IH}		0.7 x V_{CC}		$V_{CC} + 0.3$	V
Logic 0 Input SDA, SCL	V_{IL}		-0.3		0.3 x V_{CC}	V

Electrical Characteristics

($V_{CC} = 2.3V$ to $5.5V$, V_{CC} = Active Supply (see Table 1), $T_A = T_{MIN}$ to T_{MAX} , unless otherwise noted.) (Typical values are at $V_{CC} = 3.3V$, $V_{BAT} = 3.0V$, and $T_A = +25^\circ C$, unless otherwise noted.) (Notes 2, 3)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Active Supply Current	I_{CCA}	(Notes 4, 5)	$V_{CC} = 3.63V$		200	μA
			$V_{CC} = 5.5V$		300	
Standby Supply Current	I_{CCS}	I ² C bus inactive, 32kHz output on, SQW output off (Note 5)	$V_{CC} = 3.63V$		110	μA
			$V_{CC} = 5.5V$		170	
Temperature Conversion Current	$I_{CCSCONV}$	I ² C bus inactive, 32kHz output on, SQW output off	$V_{CC} = 3.63V$		575	μA
			$V_{CC} = 5.5V$		650	
Power-Fail Voltage	V_{PF}		2.45	2.575	2.70	V
Logic 0 Output, 32kHz, INT/SQW, SDA	V_{OL}	$I_{OL} = 3mA$			0.4	V
Logic 0 Output, RST	V_{OL}	$I_{OL} = 1mA$			0.4	V
Output Leakage Current 32kHz, INT/SQW, SDA	I_{LO}	Output high impedance	-1	0	+1	μA
Input Leakage SCL	I_{LI}		-1		+1	μA
RST Pin I/O Leakage	I_{OL}	RST high impedance (Note 6)	-200		+10	μA
V_{BAT} Leakage Current (V_{CC} Active)	I_{BATLKG}			25	100	nA

Electrical Characteristics (continued)

(V_{CC} = 2.3V to 5.5V, V_{CC} = Active Supply (see Table 1), T_A = T_{MIN} to T_{MAX}, unless otherwise noted.) (Typical values are at V_{CC} = 3.3V, V_{BAT} = 3.0V, and T_A = +25°C, unless otherwise noted.) (Notes 2, 3)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Output Frequency	f _{OUT}	V _{CC} = 3.3V or V _{BAT} = 3.3V		32.768		kHz
Frequency Stability vs. Temperature (Commercial)	Δf/f _{OUT}	V _{CC} = 3.3V or V _{BAT} = 3.3V, aging offset = 00h	0°C to +40°C		±2	ppm
			>+40°C to +70°C		±3.5	
Frequency Stability vs. Temperature (Industrial)	Δf/f _{OUT}	V _{CC} = 3.3V or V _{BAT} = 3.3V, aging offset = 00h	-40°C to <0°C		±3.5	ppm
			0°C to +40°C		±2	
			>+40°C to +85°C		±3.5	
Frequency Stability vs. Voltage	Δf/V			1		ppm/V
Trim Register Frequency Sensitivity per LSB	Δf/LSB	Specified at:	-40°C		0.7	ppm
			+25°C		0.1	
			+70°C		0.4	
			+85°C		0.8	
Temperature Accuracy	Temp	V _{CC} = 3.3V or V _{BAT} = 3.3V		-3	+3	°C
Crystal Aging	Δf/f _O	After reflow, not production tested	First year		±1.0	ppm
			0–10 years		±5.0	


Electrical Characteristics

(V_{CC} = 0V, V_{BAT} = 2.3V to 5.5V, T_A = T_{MIN} to T_{MAX}, unless otherwise noted.) (Note 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Active Battery Current	I _{BATA}	E _{OSC} = 0, BBSQW = 0, SCL = 400kHz (Note 5)	V _{BAT} = 3.63V		70	μA
			V _{BAT} = 5.5V		150	
Timekeeping Battery Current	I _{BATT}	E _{OSC} = 0, BBSQW = 0, EN32kHz = 1, SCL = SDA = 0V or SCL = SDA = V _{BAT} (Note 5)	V _{BAT} = 3.63V	0.84	3.0	μA
			V _{BAT} = 5.5V	1.0	3.5	
Temperature Conversion Current	I _{BATTC}	E _{OSC} = 0, BBSQW = 0, SCL = SDA = 0V or SCL = SDA = V _{BAT}	V _{BAT} = 3.63V		575	μA
			V _{BAT} = 5.5V		650	
Data-Retention Current	I _{BATTD}	E _{OSC} = 1, SCL = SDA = 0V, +25°C			100	nA

B.3 Relay 1 Channel

SONGLE RELAY

	RELAY ISO9002	SRD
---	---------------	------------



1. MAIN FEATURES

- Switching capacity available by 10A in spite of small size design for high density P.C. board mounting technique.
- UL,CUL,TUV recognized.
- Selection of plastic material for high temperature and better chemical solution performance.
- Sealed types available.
- Simple relay magnetic circuit to meet low cost of mass production.

2. APPLICATIONS

- Domestic appliance, office machine, audio, equipment, automobile, etc.
(Remote control TV receiver, monitor display, audio equipment high rushing current use application.)

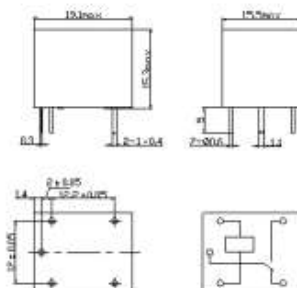
3. ORDERING INFORMATION

SRD	XX VDC	S	L	C
Model of relay	Nominal coil voltage	Structure	Coil sensitivity	Contact form
SRD	03, 05, 06, 09, 12, 24, 48VDC	S:Sealed type	L:0.36W	A:1 form A
		F:Flux free type	D:0.45W	B:1 form B
				C:1 form C

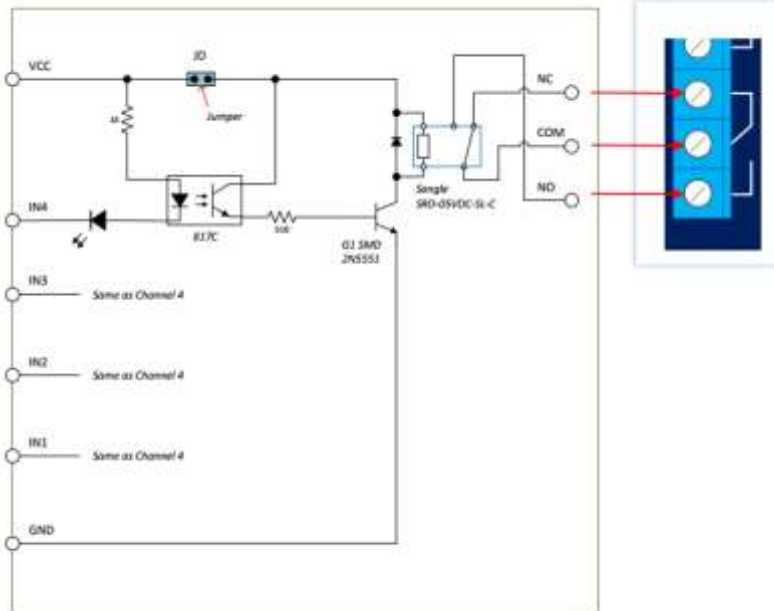
4. RATING

CCC	FILE NUMBER:CH0052885-2000	7A/240VDC
CCC	FILE NUMBER:CH0036746-99	10A/250VDC
UL/CUL	FILE NUMBER: E167996	10A/125VAC 28VDC
TUV	FILE NUMBER: R9933789	10A/240VAC 28VDC

5. DIMENSION (unit:mm) DRILLING (unit:mm) WIRING DIAGRAM

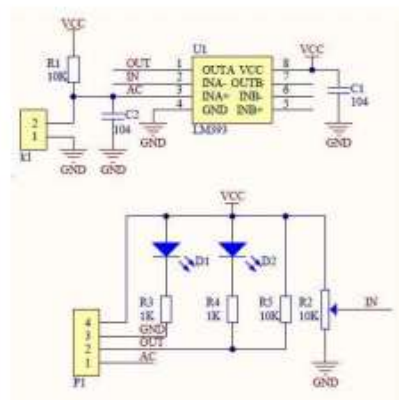


B.4 Modul Relay 4 Channel



B.5 Modul Sensor Hujan

Schematic Diagram



Specifications

- Adopts high quality of RF-04 double sided material.
- Area: 5cm x 4cm nickel plate on side,
- Anti-oxidation, anti-conductivity, with long use time;
- Comparator output signal clean waveform is good, driving ability, over 15mA;
- Potentiometer adjust the sensitivity;
- Working voltage 5V;
- Output format: Digital switching output (0 and 1) and analog voltage output AO;
- With bolt holes for easy installation;
- Small board PCB size: 3.2cm x 1.4cm;
- Uses a wide voltage LM393 comparator

Pin Configuration



1. **VCC**: 5V DC
2. **GND**: ground
3. **DO**: high/low output
4. **AO**: analog output

B.6 Blower

Specifications:

24 volts (6-24) DC

45 Watt

2000 revolutions min (Clockwise)

Air flow: 225 m / h

Sound pressure level: 64 dB (A)

Material: galvanized steel sheet

Maintenance-free ball bearings

BKV 301.216 / 51

Measurements:

Size: approx 17,1 cm x 18,0 cm x 8,1 cm

Fan opening: 11 cm (diameter)

Air outlet opening: 5,8 x 6,7 cm

Weight: about 1,3 kg

MADE IN GERMANY

D. DAFTAR RIWAYAT HIDUP



Nama : Fastabiq Khoir Alblitary
TTL : Blitar, 16 November 1996
Jenis Kelamin : Laki-laki
Agama : Islam
Alamat : Purworejo Sanankulon Blitar
Telp/HP : 085791226033
E-mail : khoirfastabiq@gmail.com

RIWAYAT PENDIDIKAN:

1. 2002-2008 : SD Islam Kota Blitar
2. 2008-2011 : MTsN 1 Blitar
3. 2011-2014 : SMK Negeri 1 Blitar
4. 2014-2017 : D3 Teknik Elektro, Program Studi
Komputer Kontrol – Fakultas Vokasi

PENGALAMAN KERJA

1. Kerja Praktek di PT Charoen Pokphand Indonesia Krian Sidoarjo

PENGALAMAN ORGANISASI

1. Panitia IARC (*Industrial Automation and Robotic Competition*)
kategori LEGO NXT 2015
2. Panitia IARC (*Industrial Automation and Robotic Competition*)
kategori LEGO NXT 2016
3. Staff Departemen Kesejahteraan Mahasiswa Himpunan Mahasiswa
D3 Teknik Elektro ITS 2015 – 2016
4. Kepala Departemen Kesejahteraan Mahasiswa Himpunan
Mahasiswa D3 Teknik Elektro ITS 2016 – 2017

